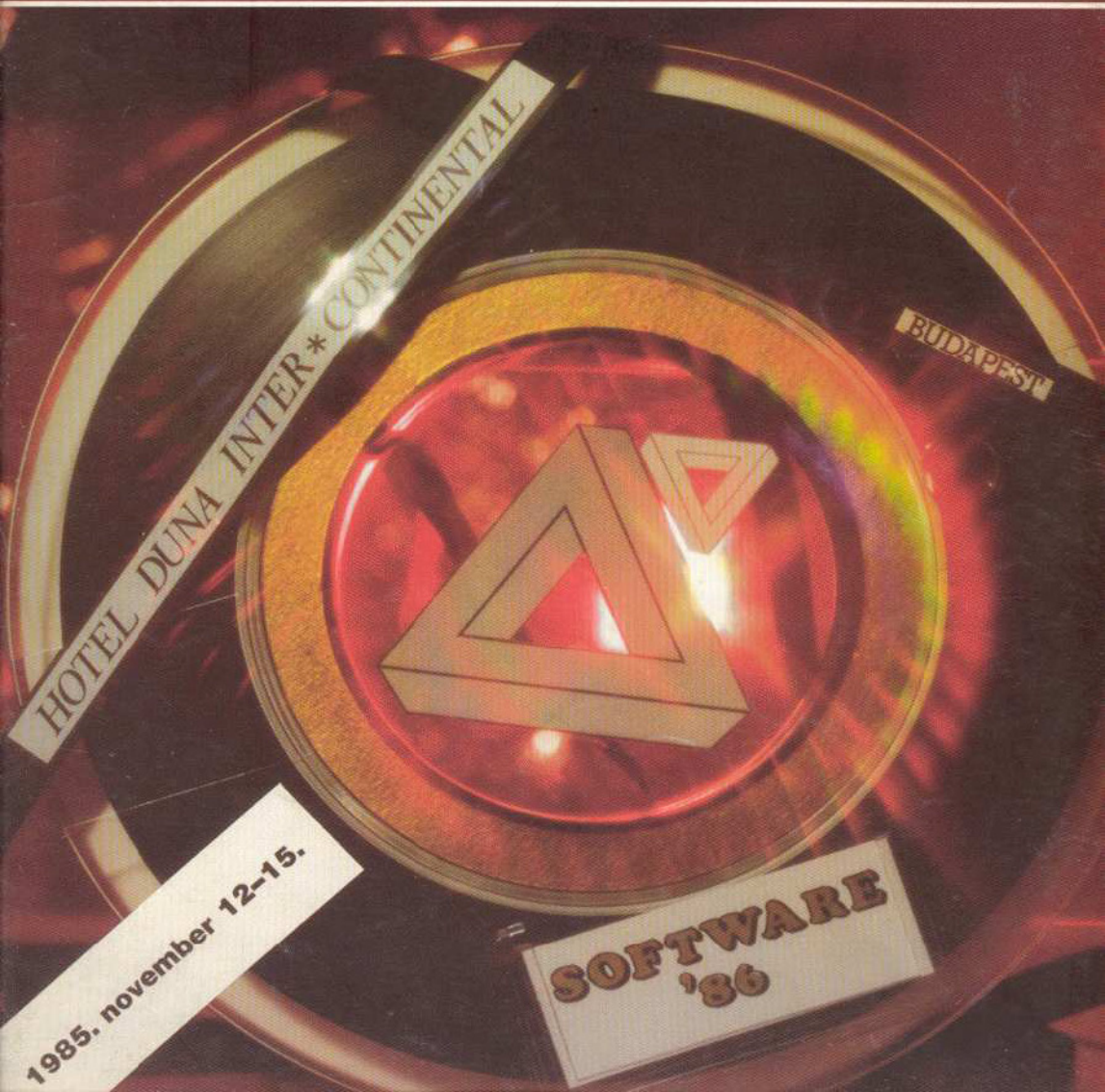
MIKROSZÁMÍTÓGÉP MAGAZIN

A NEUMANN JÁNOS SZÁMÍTÓGÉP-TUDOMÁNYI TÁRSASÁG LAPJA

1985/5



Mi talán még Önnél is többet tudunk a PC-ről!

Győződjön meg róla

a kiállításon!

– Egymunkahelyes IBM–PC kompatíbilis gépét 4 munkahelyesre bővítjük

– IBM PC, PC–XT gépekből és intelligens terminálokból Local-Network max. 128 munkahellyel

– Memóriabővítés, nyomtató-, mágnesszalag- és streamer-illesztés

– Szoftverek teljes választéka

– Alkalmazói rendszerek:

bér-, munkaügy
főkönyv-, folyószámla-nyilvántartás
anyagnyilvántartás
állóeszköz-nyilvántartás
kereskedelmi rendszerek

– PC szerviz és karbantartás (IBM–PC, PC–XT, COMMODORE PC–10, PC–20, COPAM XT 401 stb.)

Címünk: µS MICROSYSTEM SZÁMÍTÁSTECHNIKAI MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET

Budapest, XIV., Kőszeg u. 4. 1141 Tel.: 831-805

MIKROSZÁMÍTÓGÉP MAGAZIN

A Neumann János Számítógéptudományi Társaság lapja

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztő bizottság vezetője:
Kovács Győző

Munkatársak:
Broczkó Péter (hírek)
Buday György István (személyi számítógépek)
Jakab Ágnes (ember–gép kapcsolat)
Kovács Győző (levelezés)
Lindner László (sakkprogramozás)
Pataki Ernő (programozástechnika)
Petróczy Judit (könyvek)
Pogány Csaba (alkalmazástechnika, tanfolyam)
Simonyi Endre (klub)
Takácsy Ildikó (favágás)
Vadkert János (μprogramok)
Varga András (iskola – számítógép)
Vass Nándor (alkalmazások)
Votisky Zsuzsa (játékprogramok)
Zárda Sarolta (piac)

A szerkesztőség munkatársai:
Albert Tibor
Nacsa Sándor

Felelős szerkesztő:
Könyves Tóth Pál
Szerkesztőség:
1027 Budapest II., Fő utca 68.
Telefon: 154-250

Kiadja a Delta Szaklapkiadó és Műszaki Szolgáltató Leányvállalat
Felelős kiadó:
Faklen Pál igazgató
1442 Budapest, VII., Garay utca 5.
Telefon: 415-583, 215-440

Terjeszti a Magyar Posta
Előfizethető
bármely postahivatalban,
a kézbesítőknél,
a Posta hírlapüzleteiben
és a Posta
Központi Hírlap Irodában
(Budapest V., József nádor tér 1.
Postacím: 1900 Budapest)
közvetlenül
vagy postautalványon,
valamint átutalással
a PKHI 215-96162
pénzforgalmi jelzőszámra
Előfizetési díj:
egy évre 180,– Ft,
fél évre 90,– Ft.

Szedte
a Nyomdaipari Fényszedő Üzem
(857901/9)

Nyomás:
Petőfi Nyomda, Kecskemét,
Külső Szegedi út 6.
(85/50065)
Telefon: 28777
Felelős vezető:
Ablaka István igazgató

INDEX: 25629
ISSN 0236-5088

Címképünk:
Software '86
Hotel Duna Inter•Continental
1985. november 12–15.

Tartalom

Még egyszer az informatika oktatásáról	2
Software '86	3
Szövegfeldolgozók – objektum-orientáltan	9
A mikro-számítástechnika nem mikroszintet jelent!	13
Egy számítógépes fertőzés kórtörténete	16
Akik csatát veszíthetnek, de háborút nem	17
A technológiák szerepe	18
A Commodore 64 kommunikációs lehetőségei	22
Harc az európai vezető helyért	24
Adok – veszek – cserélek	29
Tanulságos kalandok a számítástechnika alkalmazásai során	30
Laser–210	44
Tudáspróba	44
ISKOLA – SZÁMÍTÓGÉP	
Diplomamunkák a számítógép iskolai felhasználásáról	4
Magyar hibaüzenetek	5
Számok kiírása a képernyőre gépi kódban	6
Ékezetes betűk nyomtatása	6
TERMÉKISMERTETŐ	
A ROSYTEXT	8
TANFOLYAM	
Alapozás XII.	10
PROGRAMOZÁSTECHNIKA	
Képletkezelő programok	14
Görbére illeszkedő polinom kiszámítása	16
μPROGRAMOK	26
μKLUB	
Építsünk számítógépet! X.	32
Kazettás nyilvántartó program	32
Klubújdonságok	35
SAKKPROGRAMOZÁS	
Lépésről lépésre	37
JÁTÉKPROGRAMOK	
Két reflexjáték	38
Űrutazás	38
Torpedó	38
AZ OLVASÓ ÍRJA	42
KÖNYVEK	45
HÍREK, ÉRDEKESSÉGEK	46

Még egyszer az informatika oktatásáról

Gépvilágosság, gépmeleg, gépzene.
A műgyerek műnyulakkal játszik.
A Nagy Számítógép rendelte így el.

És éjszakánként megszúr egy rugó,
mely asszonyunk könyökéből kiáll,
agyunkban csikorognak a fogaskerekek.

De felépítjük az emberhez méltó világot,
a Nagy Számítógép tudja, mi az.
Dicsértessék a Nagy Számítógép!

(SZENTMIHÁLYI SZABÓ PÉTER: SCIENCE-FICTION)

Az elmúlt félévben két konferencián is összejöhettek mindazok a szakemberek, akik ma számítástechnika-oktatással foglalkoznak. Egészen pontosan két – majdnem hogy különálló – témakörrel: az egyik csoport ugyanis magát a számítástechnikát (de mondják azt is, hogy informatikát, azután vannak, akik a számítástudomány kifejezést használják) tanítja, míg a másik a számítástechnika eszközeit (gépeit, programjait, sőt módszereit) alkalmazza az oktatásban. Szerény megítélésem szerint az első csoportban legfeljebb kísérletekről, ha úgy tetszik reményt keltő próbálkozásokról lehet beszélni, míg a másodikban már eredményekről. Igaz, ezeket az eredményeket sokan vitatják, talán még az eredmény voltát is kétségbe vonják.

Az egyik konferencia Várnában volt. Az előzetes megbeszélésen a Program Bizottság legfeljebb kétszáz európai résztvevőre számított, a végén a résztvevők száma megközelítette a hatszázat, nagyon sokan a tengerentúlról is.

A másik, a nevében is világkonferenciát Észak-Amerikában, Norfolkban két és félezer résztvevővel rendezték. Mindkét konferenciával párhuzamosan kiállítást is szerveztek, ahol jól lehetett látni, hogy a számítógépes oktatás ma már nemcsak törekvés, de az egyik legnagyobb üzlet is az informatikában. Az egyik amerikai kiállító azt jósolta, hogy az elkövetkező öt évben a CAI (Computer Aided Instruction) üzletek pénzügyi volumene összemérhető lesz vagy meghaladhatja bármelyik kommerciális területen kötött számítástechnikai szerződések nagyságrendjét.

Azt hiszem, nem érdemes a két konferencia mondanivalóját szétválasztani, hiszen az elhangzottak nagyjából összecsengtek. A várnai megállapításokat a norfolki felerősítette és kiegészítette a több száz amerikai, kanadai és nem utolsósorban távol-keleti szakember véleményével.

Ha az elhangzott előadásokat elemezzük és átnézzük a kb. nyolcszáz oldalas várnai és az ezeroldalas norfolki kötetet, akkor nem nehéz megállapítani, hogy szinte az egész világon a középiskolai oktatáson van a hangsúly. Ezt követi sorrendben az elemi iskolai képzés, az egyetemi, főiskolai oktatás és az elemi iskola előtti foglalkozások. Viszonylag kevesen foglalkoztak a felnőttoktatással, ezért csak néhány előadás hangzott el a tömegoktatásról, illetve a távoktatásról. (A kettőt azért kell külön említeni, mert vannak országok, ahol a tömegoktatásban a tömegkommunikációs eszközöket még nem alkalmazzák.) Az informatikai tömegoktatás fontosságát jelzi, hogy az oktatással foglalkozó IFIP technikai bizottság (TC–3) norfolki ülése támogatta a magyar javaslatot, amely szerint a technikai bizottság keretében egy, a tömegoktatással foglalkozó munkacsoport (WG) jöjjön létre. Az alakulás első lépéseként 1986 októberében nemzetközi munkakonferenciát rendezünk „Tömegoktatás és informatika" címmel.

Visszatérve a konferenciákon elhangzottakra, néhány tanulságot talán érdemes megemlíteni.

Már a bevezetőben is említettem, hogy az informatika közép- és elemi iskolai oktatását szorgalmazó szakemberek, illetve országok meglehetősen zavarban vannak; ott is hasonló a helyzet, mint nálunk. Azt senki sem vitatja ugyanis, hogy a számítástechnika tanítása hasznos dolog. A probléma akkor merül fel, ha azt kell megmondani, hogy mit is oktassanak, az már csak másodlagos kérdés, hogy még az iparilag fejlett országokban is kérdéses, hogy kikkel, mert nincs elég, az informatikában széleskörűen képzett pedagógus. Az első kérdésre, a „mit"-re gyakorlatilag sok választ adtak, de egyik sem volt elfogadható. Igen erős pártja van a programozástanításnak, de legalább ilyen erős az ellenzők tábora. Elenyészően kevesen mondták azt, hogy hardvert kell tanítani, ha mondták is hozzátették, hogy legfeljebb fogalmak megismertetéséről van szó, nem pedig mélyebb műszaki ismeretekről. Egyre erősbödni látszik a „ne tanítsunk se hardvert, se szoftvert" javaslók tábora, ők a feladatmegfogalmazásra és a problémamegoldásra helyezik a hangsúlyt. Azt mondják, hogy a diszkrét utasításos magas szintű programozási nyelvek órái megszámlálhattak, jön a logikai programozás, amelyben a feladat helyes logikai megfogalmazásán van a hangsúly és nem a program – mai értelemben vett – megírásán. „Így minden olyan perc, amit ma – ifjúsági szinten – programozás oktatására fordítanak, időpocsékolás" – mondta a vitában nem kis túlzással az egyik szakértő. Ezért volt nagy sikere és persze nagy hallgatósága is mindkét konferencián a mesterséges intelligenciáról, annak az oktatásban való alkalmazásáról, valamint a logikai programozásról, ott is elsősorban a Prologról szóló előadásoknak. Ami az elemi iskolai oktatást és az iskola előtti képzést illeti, ott – különösen Amerikában – a LOGO az uralkodó nyelv. Ha hinni lehet az előadóknak, akkor a ma alkalmazott LOGO oktatási technológiával a kicsik oktatásában egészen fantasztikus eredményeket érnek el.

A konferenciákon elég sok, a számítástechnika alkalmazása iránt érdeklődő tanár vett részt, nem kevesen előadáson ismertették, hogyan alkalmazzák a számítógépet más tantárgyak oktatására. A számítógép – én úgy láttam – egyre többször jelenik meg, mint demonstrációs eszköz. Elég sokan beszéltek arról, hogy a számítógépes animáció a lehető legjobb oktatási média pl. folyamatok ábrázolására, hiszen a folyamatba bármikor be lehet nyúlni, ha bizonyos feltételeket megváltoztatnak, a diák azonnal *látja* a változtatás hatását.

Számomra meglepő volt a jelolvasó berendezéseknek a tömeges megjelenése mind az amerikai kiállításon, mind pedig az előadások témáiban, ahol az alkalmazásukat mutatták be. Ennek az oka, hogy az iskolai számítógépeket egyre több tanár használja a beszámoltatások, vizsgák eredményének tárolására és kiértékelésére. Úgy látom, hogy megszűnőben van a „tanári notesz", miután már nagyon sok iskolában minden osztályban van számítógép, és a tanár az egyes diákokról a feljegyzéseit floppyn tárolja, majd onnan hívja le a szükséges adatokat. A beszámoltatások nagyobbik része írásos teszt, a diák a kérdés-lapot megtarthatja, csak azt a kártyát kell beadnia, amelynek a tartalmát azután a tanár a számítógéphez kapcsolt jelolvasóval olvassa el és viszi be a tárolóba. Amig az óra végén a beszámoltatással kapcsolatos gondolatait a tanár összefoglalja, addig a gép a dolgozatokat kiértékeli és az eredményt ki is nyomtatja.

Néhány jelenlévő gyakorló pedagógus azt mondta, hogy a beszámoltatás pedagógiai hatása így igen jó, mert a diák a kikérdezés után frissiben tudja, hogy hol és mit hibázott, sikeresen vizsgázott-e, vagy sem.

Mindenről – idő- és helyhiány miatt – nyilvánvalóan nem lehet beszámolni. Arról azonban érdemes néhány szót mondani, hogy szinte egyöntetű volt a vélemény, hogy az iskolákban olyan egymással kommunikáló rendszereket kell kialakítani, amelyek lokális hálózatba is szerelhetők. Ezeknek a rendszereknek jó része rá van kapcsolva az országos adathálózatra is, így a tanár és a diák a házi számítógépen is elérheti a saját vagy más iskolának a rendszerét. Régi szabály, hogy az iskola az életre készíti fel a diákokat, így a számítástechnikában is a külső gyakorlatot kell modellezni. Ezért hoznak létre – szinte kivétel nélkül a fejlett országokban – iskolai hálózatokat, az önálló, egymással összeszerelhető iskolai gépek helyett.

A hazai iskolai informatikai oktatás ma világmércével mérve is megállja a helyét. Kiegészítve és továbbfejlesztve a tömegoktatási rendszerrel, hasznosan szolgálja a társadalom informatizálását. Nagy hiba lenne, ha ennyivel megelégednénk, tovább kell lépnünk. Iskolai rendszereket, hálózatokat kell kialakítanunk, tovább kell növelni az alkalmazói programok mennyiségét, meg kell tenni az első lépéseket a logikai programozás terjesztésére, végül meg kell teremteni a távoktatás státusát a hazai oktatási rendszerben.

KOVÁCS GYŐZŐ

SZOFTVER SZOFTVER ÁRU? SOFTWARE '86

Ha azt a szót halljuk: szoftver, már nem nyúlunk szótárhoz, nem ingatjuk kérdőn fejünket – természetesnek vesszük. Itt van, velünk él, belopta magát mindennapjainkba. Nem feltűnő, ha újságot lapozva különböző gmk-k, pjt-k, szövetkezetek, szoftverházak hirdetik magukat és termékeiket. PC-k és „mikro"-k árasztanak el minket otthon és a munkahelyünkön, barátaink, ismerőseink tanácsot kérnek, „programot fejtenek", kazettát cserélnek. Még a televízió is programozni, gépet kezelni tanít.

És tudomásul vesszük, hogy gyermekeink „új világ"-ban élnek, új matematikát tanulnak, „számítógépes" nyelven beszélnek; mi még csak pedzegetjük, izlelgetjük, hogy bit meg bájt, ők már arról vitatkoznak, hogy MVC vagy LOAD.

Így aztán azon sem csodálkozunk, ha egy kiállításon vagy vásáron, ahol a számítástechnika is jelen van, komolyan „játszó" vagy programozó gyerekek hada lepi el a gépeket, kiszorítva az érdeklődő felnőtteket. A BNV-n megtalálják a számítógépeket – bárhol vannak is – és a játékprogramokat, de még inkább jelen vannak egy csak számítástechnikával foglalkozó rendezvényen. Ez így van rendjén: ők a jövő emberei, bennük a lehetőség, hogy a számítástechnikát természetes eszközként használják és alkalmazzák felnőtt korukban, egy technikailag még fejlettebb környezetben.

Mindezeket a gondolatokat az ez évi legnagyobb számítástechnikai kiállítás juttathatja eszünkbe.

Novemberben ismét találkozhatnak a számítástechnika iránt érdeklődők a HOTEL DUNA INTER*CONTINENTAL-ban, ahol a COMPORGAN RENDSZERHÁZ K.V. rendezi a SOFTWARE '86-ot, a szoftver áruk országos seregszemléjét. Több mint 40 kiállító – köztük néhány külföldi – mutatkozik be, mikroszámítógépekkel és más eszközökkel felszerelkezve, hogy újra bizonyítékot adjanak munkájukról, eredményeikről, fejlesztési elképzeléseikről.

A két évvel ezelőtt rendezett SOFTWARE '84 kiállításhoz hasonlóan színvonalas, egységes köntösű standokon jelennek meg a különböző vállalatok, intézmények, így is jelezve: termékük az elsődleges, a fontos, ezzel kívánnak kitűnni a sorból.

A kiállítás jó alkalom arra, hogy – az alapvető cél: a látogatók, az érdeklődők tájékoztatása mellett – a különböző szoftvergyártó és forgalmazó vállalatok, gmk-k, szövetkezetek egymás munkáit is jobban megismerjék. Ez lehetőséget biztosíthat számukra az újszerű témaválasztáshoz, a gazdaságos fejlesztéshez.

Természetesen itt is tanúi leszünk annak, hogy hasonló hazai termékek egymásra köszönnek, de reméljük, ebből a kiállítás látogatói húzzák majd a legnagyobb hasznot. A helyszínen akár azonnal összehasonlíthatják azokat, és dönthetnek az egyik vagy másik javára. Bízzunk benne, hogy mindezek az esetek az egészséges versenyszellem kialakulását szorgalmazzák.

Biztosak lehetünk abban is, hogy ismét számos gyereklátogató tölti majd itt a négy napját. A nekik és tanáraiknak egy éve meghirdetett és azóta folyamatosan tartó ÖTLETPÁLYÁZAT eredményhirdetésére is a SOFTWARE '86 rendezvényeként kerül sor 1985. november 13-án, szerdán délelőtt 11 órakor.

A számítástechnikát először alkalmazni kívánóktól a potenciális vásárlókig minden érdeklődőt egy újszerű rendezvényre is invitál a rendezőség a kiállítás alkalmából.

A SOFTWARE '86 kiállítás mellett a COMPORGAN szoftveráru-vásárt is rendez. A HOTEL DUNA INTER*CONTINENTAL egyik termében mikrogépeket helyeznek el, és helyszíni bemutatás kíséretében árusítják azt a max. 50 szoftvertermékет, amelyeket – a beküldött több, mint 100 pályázat közül – egy erre hivatott bíráló bizottság választ ki. Magát a szoftvervásárt az Ipari Minisztérium támogatja, és a kiválasztott áruk közül a legjobbnak ítélt hármat díjazásban részesíti:

I. díj: 100 000,— Ft
II. díj: 75 000,— Ft
III. díj: 50 000,— Ft

A vásár részvételi feltételei és a bíráló bizottság értékelése komoly referenciát ad a kiválasztott szoftvertermékнek, nagy elismerést jelent fejlesztőjének.

Mindezek ismeretében nézzünk várakozással a SOFTWARE '86, a Szoftver áruk kiállítása és vására elébe. Bízzunk abban, hogy valamenynyi látogató talál számára is érdekes, új dolgot a kiállításon, és gyermekeink megint gazdagodhatnak újabb, kedvcsináló ismeretekkel.

SZOFTVER ÁRUK KIÁLLÍTÁSA ÉS VÁSÁRA

HOTEL DUNA INTER*CONTINENTAL

1985. november 12–15.

Nyitva naponta: 10–18 óráig
Nyitás: 1985. november 12-én 12 órakor

Érdemes-e a

kiállításon felkeresni a

KRONOSZ

Számítástechnikai Társaságot?

Ha Ön Commodore 64-re keres általános ügyviteli programokat, akkor feltétlenül, hiszen KATALÓGUS, KORREKTOR, KALKULÁTOR programjainkat a helyszínen átveheti, munkahelyén kipróbálhatja, s csak ezután kell eldöntenie, hogy megveszi-e vagy nem.

Ha Ön nem keres ilyen programokat, mert sikerült néhány minőségi fájlkezelőt, szövegszerkesztőt, táblázatkezelőt beszereznie, akkor sem árt megkeresni a KRONOSZ-t; védett lemezeiről – az esetleges sérülések megelőzésére – a helyszínen biztonsági másolatot készítünk.

Ha Ön nem szereti a konfekciót, hanem testreszabott programokat keres, akkor mindenképp szükség van a személyes találkozóra, másképp lehetetlen mértéket venni. Az ügyviteli problémák közül különösen nagy gyakorlatunk van a bérelszámolás számítógépesítésében, a különféle nyilvántartási feladatokban, optimalizálási témákban.

Ha mindezek ellenére Ön úgy érzi, hogy nem érdemes a kiállítás szűk négy napjában felkeresni a KRONOSZ-t, akkor ne tegyen egy lépést se! Üljön le, és írja meg levélben szoftvergondját a következő címre:

KRONOSZ

Számítástechnikai Társaság
8001 Székesfehérvár, Pf. 150.
Tel.: (22)-15717

Diplomamunkák a számítógép iskolai felhasználásáról

A lapunkhoz érkezett több cikkből úgy tűnik, mintha a számítástechnika a matematikaoktatás kiegészítőjeként találta volna meg a helyét az iskolában, és minden más irányú kapcsolata csak alárendelt szerepet játszana ehhez képest. Véleményünk szerint ez éles ellentmondásban van mind az iskolákban kitűzött céllal, mind a számítástechnika valóságával, amit egy sajátos műszaki ismeretrendszerként lehet leginkább jellemezni. A számítástechnikai eszközök nyújtotta lehetőségeket elsősorban a gépek és perifériális eszközök műszaki jellemzői határozzák meg. Az alkalmazott algoritmusok java része sem az alkalmazott matematika numerikus módszerein alapszik. A programozási és programfejlesztési készséget is egy külön diszciplína, az ún. software engineering ismeretrendszerében lehet elsajátítani. A számítástechnikai alapismeretek elsajátításához egy sor szoftverrendszerrel, mint operációs rendszer, programozási nyelvek, adatbáziskezelők stb. is meg kell ismerkedni. Ezért a jövőben szívesen helyet adnánk olyan írásoknak, amelyek ilyen aspektusból kívánják segíteni a számítástechnika iskolai oktatását.

A személyi számítógépek megjelenése a középiskolákban nemcsak az ottani életben hozott lényeges változást, de érezteti hatását az egyetemen a matematika-fizika szakos tanárjelöltek oktatásában is.

Megfigyelhető ez azon is, hogy az utóbbi években ugrásszerűen megnőtt azoknak a szakdolgozatoknak a száma, amelyek a személyi számítógépek iskolai felhasználásával foglalkoznak. Az egyetemi tanulmányok utolsó évében ti. minden egyetemistának szakdolgozatot – diplomamunkát – kell írnia egy általa választott témából. A szakdolgozat megírása és elfogadása feltétele az abszolutórium megszerzésének, ami azt jelenti, hogy a hallgató államvizsgára jelentkezhet. Ezenkívül a szakdolgozatra kapott érdemjegy beszámít az államvizsga átlagába is.

A továbbiakban az idei „termésből" mutatunk be két munkát. Ezeket ötödéves matematika-fizika-számítástechnika szakos tanárjelöltek készítették.

Gimnáziumi számítástechnika szakkör

Takács Attila fenti című szakdolgozatában 3 éves munkájának tapasztalatait, eredményeit írja le. A Jedlik Ányos Gimnázium tanulóinak már 1982-ben tartott egy féléves számítástechnikai szakkört. Mivel akkor még az iskolában nem volt számítógép, az ELTE ABC–80 személyi számítógépeit használták. Az itt szerzett tapasztalatok sok segítséget jelentettek a következő tanévben beinduló számítástechnika szabadsávhoz.

Még fél évet kellett várni, amíg sikerült egy Aircomp–16-ot vásárolniuk. A következő két év alatt számítástechnika kabinetet alakítottak az iskolában, amely már 2 db Aircomp–16, 3 db HT–1080Z és 4 db ZX-Spectrum géppel van felszerelve. Ez a kabinet jelentősen megkönnyítette a tanárok és diákok munkáját. Ügyesen kihasználták azt, hogy többfajta számítógéppel dolgozhattak egyszerre. A tanulók így megismerték a BASIC nyelv általánosan használható részét, és közben az egyes gépek specialitásait is. Nagyszerűen be lehetett mutatni, hogy a különböző problémák megoldására hol az egyik, hol a másik típus használható egyszerűbben, előnyösebben. A másodikosok részére tartott foglalkozások tematikája a következő volt:

1. foglalkozás: Ismerkedés a számítógéppel, felhasználási területeivel. Néhány számítógépes játék.

2. foglalkozás: Az algoritmus és a program fogalma. A kétféle üzemmód. Az értékadás és értékek beolvasása, kíiratása.

3. foglalkozás: Feltételes és feltétel nélküli vezérlésátadás. A program futásának megszakítása, megállítása.

4. foglalkozás: A programok tárolása kazettán.

5. foglalkozás: A ciklusok. A ciklusszervezés feltételes utasítás segítségével, illetve a ciklusutasítás felhasználásával. Elöl és hátul tesztelő ciklusok.

6. foglalkozás: A ciklusszervezés gyakorlása. A számelmélettel kapcsolatos programok. A BASIC-ben használható legfontosabb függvények.

7. foglalkozás: A képernyőgrafika. A különböző gépek grafikai lehetőségeinek összehasonlítása.

8. foglalkozás: További grafikai feladatok. Függvények ábrázolása.

9. foglalkozás: A véletlenszám-generátor.

10. foglalkozás: A karakteres változók. Konkatenáció. Szubrutinok.

11. foglalkozás: Műveletek karakterláncokkal, a karakterkezelő függvények. ASC kódok.

12. foglalkozás: Tömbök.

13. foglalkozás: A hanggenerátor.

Számítógép egy középiskolai matematika szakkörön

Mátyás Andrea szakdolgozatát a Szerb Antal Gimnáziumban tartott egyéves matematika szakköréről írta.

A szakkörre jelentkező tanulók ugyan különböző szinten, de már rendelkeztek programozási ismeretekkel. Így lehetőség volt arra, hogy egy év alatt a matematika számos olyan fejezetével foglalkozzanak, amelyek feldolgozásához a számítógép jól használható. Amikor olyan matematikai ismeret került elő, amit a szakkör résztvevői közül nem mindenki ismert, akkor a felsőbb évesek lelkesen magyaráztak a többieknek. Hasonló volt a helyzet akkor is, amikor ügyes programozási fogásokat, szellemes megoldásokat ismertek vagy találtak a diákok. A tanulók iskolaszámítógépen és ZX-Spectrumon dolgoztak.

A 24 foglalkozás gazdag anyagának érdekesebb témái a következők voltak:

- Oszthatósággal kapcsolatos programok (egy adott szám osztója-e egy másik adott számnak, egy adott szám osztóinak meghatározása, két szám legnagyobb közös osztójának és legkisebb közös többszörösének megkeresése stb.).
- Számsorozatokkal kapcsolatos problémák. Pozitív valós szám közelítése racionális törtszámokkal. A sorozat megadása rekurzióval. A helyettesítési értékek számolása különböző módokon. A Newton-féle gyökvonó algoritmus. Az „e" és a „π" szám különböző közelítései.
- A számítástechnika története.
- Ismerkedés a Primóval.
- Egyenletek közelítő gyökeinek meghatározása különböző módszerekkel (intervallumfelezés, húrmódszer, fixpont-iteráció).
- Mátrixok és determinánsok.
- A véletlenszám-generátor és az elemi valószínűségszámítás. Geometriai valószínűség.
- Különböző sorba rendezési eljárások (buborék-algoritmus, Shell-módszer, Quick sort módszer).
- A gépi kódú programozás alapjai.

Ezek a szakdolgozatok garanciát jelentenek arra, hogy készítőik a továbbiakban is hatékonyan fogják a számítógépet tanári munkájukban felhasználni, színesebbé, tartalmasabbá téve ezzel a matematikaórákat.

MAJOR ZOLTÁN

TV-BASIC NÍVÓDÍJ

A Magyar Televízió félévenként értékeli a közvetített műsorok társadalmi hasznosságát, jelentőségét és színvonalát. A TV-BASIC tanfolyamot korrekt, világos, jól feldolgozott műsornak ítélte, és ezért a belső és külső közreműködőket nívódíjban részesítette.

A belső munkatársak közül nívódíjat kapott Albert József rendező forgatókönyvíró, Hegyi István szerkesztő forgatókönyvíró, Ambrusán János műszaki vezető, Csillag Zoltán gyártásvezető, Váraljai Sándor felvételvezető, Fejes Csaba rögzítésvezető, Galambos György világosító, Kovács Gabriella a rendező munkatársa, Dobay Sándor vezető operatőr, Herbert István hangmérnök, Dobos Menyhért képmérnök, Fekete Lajos képvágó, Schmidt István fővilágosító.

A műsor készítésében közreműködő külső munkatársak: Kovács Győző az akció előkészítéséért, támogatásáért és előadói munkájáért, Herneczki Katalin, Kiss Donát, dr. Kocsis András és Pinkert László pedig előadói munkájukért szintén részesültek a díjban.

HT-1080Z

Magyar hibaüzenetek

Kezdő programozókkal gyakran előfordul, hogy a program valamilyen hibajelzéssel megáll. Célszerűnek látszik egy olyan program megírása, amely a kétbetűs angol hibajelek helyett teljes mondatos, magyar hibaüzeneteket produkál.

Ehhez kissé bele kell nyúlni az interpreter lelkivilágába, amit csak gépi programmal tehetünk meg. Az alábbi gépi program BASIC-ben POKE utasításokkal tölthető be a memóriába. Miután ez a BASIC program lefut, kitörlődik, de ekkor már működik az általa generált gépi szubrutin. Ezt nem tudjuk kitörölni sem a RESET gombbal, sem a NEW paranccsal; a gép kikapcsolásáig küldi a magyar hibaüzeneteket.

GAUTIER PÉTER

```
                ;-------------------------------------------------
                ;----            EKEZET KISZEDO PROGRAM
                ;----            GAUTIER PETER 1984.11.04.    KSZ
                ;-------------------------------------------------
                ;
      40A4      BASPR   EQU   40A4H
      0093      REM     EQU   93H
      00FB      APOSZ   EQU   0FBH
                ;
      7F20              ORG   7F20H
                ;
7F20 ED5BA440   START:  LD    DE,(BASPR)
7F24 13                 INC   DE
7F25 13                 INC   DE
7F26 13                 INC   DE
7F27 13                 INC   DE
7F28 1A         CIKL:   LD    A,(DE)
7F29 FE22               CP    '"'
7F2B CC667F             CALL  Z,EKEZET
7F2E FEFB               CP    APOSZ
7F30 CC437F             CALL  Z,COMM
7F33 FE93               CP    REM
7F35 CC437F             CALL  Z,COMM
7F38 B7                 OR    A
7F39 2005               JR    NZ,LE
7F3B CD5C7F             CALL  VEGE
7F3E 13                 INC   DE
7F3F 13                 INC   DE
7F40 13         LE:     INC   DE
7F41 18E5               JR    CIKL
                ;
                ;REM VAGY ' ESETEN
7F43 13         COMM:   INC   DE
7F44 1A                 LD    A,(DE)
7F45 B7                 OR    A
                .
                .
                .
7F46 C8                 RET   Z                   ;SOR VEGE
7F47 CD4C7F             CALL  EKES
7F4A 18F7               JR    COMM
                ;
                ;HA EKEZETES BETU, KICSERELI
7F4C 0608       EKES:   LD    B,8
7F4E 21727F             LD    HL,TABLA
7F51 BE         LEB:    CP    (HL)
7F52 23                 INC   HL
7F53 2804               JR    Z,LE2               ;EKEZETES
7F55 23                 INC   HL
7F56 10F9               DJNZ  LEB
7F58 C9                 RET
7F59 7E         LE2:    LD    A,(HL)
7F5A 12                 LD    (DE),A
7F5B C9                 RET
                ;
                ;MEGVIZSGALJA, HOGY CSAK SOR VAGY PROGRAM VEGE
7F5C 13         VEGE:   INC   DE
7F5D 1A                 LD    A,(DE)
7F5E B7                 OR    A
7F5F 13                 INC   DE
7F60 C0                 RET   NZ
7F61 1A                 LD    A,(DE)
7F62 B7                 OR    A
7F63 C0                 RET   NZ
7F64 F1                 POP   AF
7F65 C9                 RET                       ;VISSZA A BASICB
                ;
                ;STRINGET VIZSGAL
7F66 13         EKEZET: INC   DE
7F67 1A                 LD    A,(DE)
7F68 FE22               CP    '"'

7F6A C8                 RET   Z                   ;VEGE A STRINGNE
7F6B B7                 OR    A
7F6C C8                 RET   Z                   ;VEGE A SORNAK
7F6D CD4C7F             CALL  EKES
7F70 18F4               JR    EKEZET
                ;
                ;A TABLAZAT TARTALMAZZA, HOGY MIBOL MI LESZ
7F72 5D         TABLA:  DEFB  5DH
7F73 41                 DEFM  'A'
7F74 7D                 DEFB  7DH
7F75 61                 DEFM  'a'
7F76 40                 DEFB  40H
7F77 45                 DEFM  'E'
7F78 60                 DEFB  60H
7F79 65                 DEFM  'e'
7F7A 5C                 DEFB  5CH
7F7B 4F                 DEFM  'O'
7F7C 7C                 DEFB  7CH
7F7D 6F                 DEFM  'o'
7F7E 5E                 DEFB  5EH
7F7F 55                 DEFM  'U'
7F80 7E                 DEFB  7EH
7F81 75                 DEFM  'u'
7F82 00                 DEFB  0,0
7F83 00
      0000              END
```

```
10 '-------------------------------------------------------------
20 '---            Ékezet kiirto program                      ---
30 '--- Készitette: Gautier Péter    1984.11.04.   KSZK       ---
40 '-------------------------------------------------------------
50 IF PEEK(32936)=201 THEN 320 :'--- Uj gép ---
60 CLEAR 300
70 A$=STRING$(256,32)
80 POKE 16414,71 : POKE 16415,49 :'-- Kisbetük bekapcsolása -
90 POKE 16526,32 : POKE 16527,127 :'--- Gépi rutin cime 7F20 --
100 CLS : PRINT@8*64+20,"Dolgozom!  Ne zavarj!";
110 C=7*4096+15*256+32 :'--- A programot ide pakoljuk le ---
120 BN=180 :'--- Byte szám ---
130 GOSUB 180
140 A=USR(0) :'--- A bövités elinditása ---
150 CLEAR 50
160 GOTO 320
170 '--- Hexa byte-ot lerako szubrutin ---
180 FOR I=1 TO BN
190 READ B$
200 B1=ASC(LEFT$(B$,1)) : B2=ASC(RIGHT$(B$,1))
210 IF B1>ASC("9") THEN B1=B1-ASC("A")+10 ELSE B1=B1-ASC("0")
220 IF B2>ASC("9") THEN B2=B2-ASC("A")+10 ELSE B2=B2-ASC("0")
230 B=16*B1+B2 : POKE C,B : C=C+1
240 NEXT I
250 RETURN
260 DATA ED,5B,A4,40,13,13,13,13,1A,FE,22,CC,66,7F,FE,FB,CC,43
270 DATA 7F,FE,93,CC,43,7F,B7,20,05,CD,5C,7F,13,13,13,18,E5,13
280 DATA 1A,B7,C8,CD,4C,7F,10,F7,06,08,21,72,7F,BE,23,28,04,23
290 DATA 10,F9,C9,7E,12,C9,13,1A,B7,13,C8,1A,B7,C8,FE,C9,13,1A
300 DATA FE,22,C8,B7,C8,CD,4C,7F,18,F4,5D,41,7D,61,40,45,60,65
310 DATA 5C,4F,7C,6F,5E,55,7E,75,00,00
320 '--- Itt folytatodna a program ---
330 STOP
```


TUDOMÁNY

Az 1985. szeptemberétől magyar nyelven is megjelenő Scientific American 140 éve tudósít rendszeresen az elméleti vagy alkalmazott természettudományok legújabb eredményeiről. Tudományos magazinként elsősorban az érdeklődő laikusokhoz szól. A cikkeket viszont maguk a kutatók írják, elmélyült együttműködésben a lap szerkesztőivel. A Tudomány néven havonta megjelenő magyar kiadás az eredetihez hasonló, nívós kivitelben készül. Példányonkénti ára: 98,– Ft.

Számok kiírása a képernyőre gépi kódban

Gépi kódú programok írása során gyakran előfordulhat, hogy adatokat kell kiírni a képernyőre. Ez azt jelenti, hogy bináris számokat decimálissá, aztán pedig sztringgé kell alakítani. Ez nem könnyű feladat, de fölösleges is birkózni vele, mert a HT–1080Z számítógép ROM-jában rendelkezésünkre állnak a megfelelő szubrutinok.

Az alábbiakban az egész számok képernyőre vitelének két módszerét ismertetjük. Az egész számok két bájtban (16 biten) vannak ábrázolva. A Z80 sajátosságainak megfelelően a kisebb memóriacímen, tehát elöl áll az alacsonyabb helyi értékű bájt (LSB).

A legegyszerűbb módszer a ØFAFH szubrutin meghívása. Ez a HL regiszterpár tartalmát írja ki tízes számrendszerben a képernyőre a 4020H–4021H bájtokban megjelölt helytől kezdődően (kurzorpozíció). Ide a szubrutin meghívása előtt le kell tárolni egy 3CØØH–3FFFH intervallumba eső címet (képernyő-memória).

A következő programrészlet a 7ØØØH–7ØØ1H és a 7ØØ2H–7ØØ3H bájtokban levő egész számokat írja ki a képernyőre az első és a második sor közepétől kezdve:

```
LD    SP,75ØØH      Verem a visszatérési címek tárolására
LD    A,ØFH         A kurzor letiltásának kódja és
CALL  ØØ33H         a végrehajtó rutin meghívása
CALL  Ø1C9H         A képernyő törlése
LD    HL,3C2ØH      Az első sor közepének a címe
LD    (4020H),HL    A kurzorpozíció kijelölése
LD    HL,(7ØØØH)    Az első szám átvitele a HL-be
CALL  ØFAFH         Kiírás a képernyőre
LD    HL,3C6ØH      A második sor közepének a címe
LD    (4020H),HL    A kurzorpozíció kijelölése
LD    HL,(7ØØ2H)    A második szám átvitele a HL-be
CALL  ØFAFH         Kiírás a képernyőre
...
```

A szubrutinból való visszatérés után a kurzor a szám utáni első pozícióba mutat, amire figyelni kell, ha több számot akarunk egymás mellé írni.

A program a számok kiírását a megadott pozíciótól kezdi, így táblázat készítésénél például a megfelelő helyi értékek nem kerülnek egymás alá. Ezt a problémát a 132FH kezdőcímű szubrutin meghívásával oldhatjuk meg.

Ez a szubrutin a 4121H–4122H bájtokban (az ún. munkaregiszterben) levő bináris egész számot átalakítja sztringgé és letárolja a HL által mutatott címtől kezdődően. A sztringet egy ØØH bájttal zárja le. Ha a HL a képernyő-memóriába mutat, akkor a szám felkerül a képernyőre. Ebben az esetben a sztring végét mutató ØØH bájtot fel kell cserélni a szóköz 2ØH kódjára.

A következő program a 7ØØØH–7ØØ1H és a 7ØØ2H–7ØØ3H bájtokban levő egész számokat írja ki a képernyő első és második sorának közepére:

```
LD    SP,75ØØH      A program eleje megegyezik az előzővel
LD    A,ØFH
CALL  ØØ33H
CALL  Ø1C9H
LD    BC,ØØH        A számok kiírásának előkészítése
LD    HL,(7ØØØH)    Az első szám átvitele a
LD    (4121H),HL    munkaregiszterbe
LD    HL,3C2ØH      A képernyő első sorának a közepe
CALL  132FH         A szám kiírása a képernyőre
LD    (HL),2ØH      A ØØH bájt törlése
LD    BC,ØØH        A második szám kiírásának előkészítése
LD    HL,(7ØØ2H)    A második szám átvitele a
LD    (4121H),HL    munkaregiszterbe
LD    HL,3C6ØH      A második sor közepe
CALL  132FH         A második szám kiírása a képernyőre
LD    (HL),2ØH      A ØØH bájt törlése
...
```

A programot futtatva láthatjuk, hogy a számok a helyi értékek szerint kerültek egymás alá, s az ötnél kevesebb jegyű számoknál az elöl levő üres helyeken nulla áll.

A BC regiszterpár módot nyújt arra, hogy hármasával tagoljuk a számjegyeket, és tizedespontot is használjunk. A C regiszter által mutatott pozícióba és onnan kezdve 3 számjegyenként vessző kerül, a B regiszter által mutatott helyre pedig tizedespont. Próbáljuk ki például a BC = Ø5Ø2H esetet.

Ezekben a programokban még két szubrutint használtunk. A Ø1C9H címen kezdődő rutin törli a képernyőt. A ØØ33H címen kezdődő rutin kiírja a képernyőre az A regiszterben levő karaktert, de tudja értelmezni a vezérlőkaraktereket is (lásd például az iskolaszámítógép BASIC kézikönyvének 90. oldalán).

Megjegyezzük, hogy e szubrutinok általában megváltoztatják a regiszter tartalmát, amire használatuk során ügyelni kell. JUHÁSZ TIBOR

Ékezetes betűk nyomtatása

Akik már nyomtattak ékezetes betűket is tartalmazó szöveget TMT–120 vagy más típusú mátrixnyomtatóval, azok már találkoztak azzal a problémával, hogy más karakterek jelentek meg a papíron, mint a képernyőn. Ennek kiküszöbölésére szolgál az alábbi, HT–1080Z-re készült program.

```
                        ; KESZITETTE: KISPAL ISTVAN
                        ; DUNAUJVAROS
                        ; BANKI DONAT SZAKKOZEPISKOLA
                        ; PRINTER DCB MODOSITASA
                              ORG  4026H
                              LOAD 4026H
 4026 5440                    DB   54H,40H
                        ; KOD KONVERTALAS
                              ORG  4054H
                              LOAD 4054H
 4054 D5                      PUSH DE
 4055 79                      LD   A,C
 4056 FE5F                    CP   95
 4058 3802                    JR   C,TOVA1
 405A D620                    SUB  20H
 405C FE40              TOVA1: CP   64
 405E 2004                    JR   NZ,TOVA2
 4060 3E5B                    LD   A,91
 4062 180E                    JR   VIZSG
 4064 FE5D              TOVA2: CP   93
 4066 2004                    JR   NZ,TOVA3
 4068 3E40                    LD   A,64
 406A 1806                    JR   VIZSG
 406C FE5E              TOVA3: CP   94
 406E 2002                    JR   NZ,VIZSG
 4070 3E5D                    LD   A,93
 4072 5F                VIZSG: LD   E,A
 4073 79                      LD   A,C
 4074 FE5F                    CP   95
 4076 7B                      LD   A,E
 4077 3802                    JR   C,VEGE
 4079 D620                    ADD  A,20H
 407B 4F                VEGE: LD   C,A
 407C D1                      POP  DE
 407D C38E05                  JP   58EH
                              END
```

A program első része elhelyezi a nyomtató DCB-jében a 4026H–4027H címekre a karakterkódok módosítására szolgáló főprogram kezdőcímét. A program begépelése után az assemblálást célszerű kazettára végeztetni a kétszeri ORG-kijelölés miatt.

A tárgyprogram memóriavédelmet nem kíván, mivel a lemezegység részére fenntartott tárterületen helyezkedik el. Ebből viszont az következik, hogy csak lemezegység nélküli számítógépnél alkalmazható, vagy más tárterületre kell áthelyezni.

A használat során a program kazettáról történő betöltése, majd a BREAK billentyű lenyomása után a számítógép READY felirattal visszatér a BASIC-hez, és az ékezetes betűk nyomtatására kész. A gép kikapcsolásáig (még a RESET gomb használata esetén is) a program működőképes marad.

Ez a megoldás a TMT–120 típusú mátrixnyomtatót illeszti az iskolaszámítógéphez, mivel annak ékezetes betűi karakterkódját tartalmazza. Értelemszerűen e kódok átírásával más típusú nyomtatót is illeszthetünk a HT–1080Z számítógéphez. KISPÁL ISTVÁN

SW.'86

SZERZŐDÉSES EGYSÉGEK INFORMÁCIÓS RENDSZERE

A RENDSZER CÉLJA:
a vállalatok és szövetkezetek szerződéses üzemeltetési egységeinek elszámoltatása.

FUNKCIÓI:

A szerződés adatainak rögzítése, lekérdezése és módosítása. Alapbizonylat: a szerződés. A programcsomag tudja kezelni a fix havi átalánydíjas szerződések mellett a szezonális (téli-nyári) átalánydíjas szerződéseket is.

A szerződésekhez záradékot is készíthet a rendszer.

Egy diszken 3 bolt adatai férnek el.

Hóközi adatfelvitel, mely a tárgyhó folyamán bekövetkező pénzmozgásokat kíséri figyelemmel.
Likvidáció, mely a tárgyidőszak árumozgását követi.
Havi zárás, mely során az alábbi listák készülnek:

1. Főkönyvi feladás
2. ELABA-számítás
3. Tárgyhavi csekkek
4. Összesítő likvidáció
5. Közületi eladások

A PROGRAMCSOMAG ÁRA: 20 500,— Ft
KÉSZÍTETTE: UNIÓ ÁFÉSZ (MISKOLC)

Szükséges konfiguráció:

COMMODORE 64
1 vagy 2 floppymeghajtó
1 nyomtató
1 tv-készülék

FORGALMAZZA:

KERESKEDELMI SZERVEZÉSI INTÉZET

KERSZI

ÉRDEKLŐDNI LEHET:

1134 Budapest,
Dózsa György út 150.
MARKETING OSZTÁLY
Kontur Istvánné
202-650/165

A ROSYTEXT

A tavaszi BNV-n kiválóan szervezett bemutató hívta fel a figyelmet a ROLITRON Műszaki-Fejlesztő Kisszövetkezet szövegszerkesztő és szövegfeldolgozó berendezésére. A sajtóban jól megfogalmazott és elhelyezett hirdetések ostromolják a potenciális vevőket. A dinamikus és példamutatóan menedzselt kisvállalat a lehető leghatározottabb elképzelésekkel próbálja kialakítani vevőkörét, és tudja, hogy a növekedésnek aligha van nagyobb erőforrása, mint a minden szempontból elégedett ügyfelek egyre bővülő tábora. A cég józan határozottságát mi sem mutatja jobban, mint prospektusának kísérő szövege: „Ha könnyedén tudja növelni a gépírók létszámát, vagy olyan szerencsés helyzetben van, hogy munkatársai között több gépíróbajnok is van, és emiatt a jelenlegi adminisztrációs helyzetével elégedett, akkor e kiadványunkat egyszerűen dobja a szemétkosárba!"

Személyi számítógépes alap

A ROSYTEXT rendszerek a ROSY–80 személyi-számítógép-család F és MF típusjelzésű modelljein alapulnak. Az előbbinél 1 db 8 hüvelykes, az utóbbinál 2 db 5 1/4 hüvelykes floppy háttértároló van beépítve a képernyős készülékházba. Mi olyan rendszereket láttunk, amelyekbe a nagy floppy volt beépítve. A képernyő alapszíne szemet kímélő zöld, és a megjelenítés minősége kiváló. Nem tapasztaltuk például a más gépeknél elég sajnálatos módon előforduló „sorimbolygásokat" és más kellemetlen képernyőeffektusokat.

A billentyűzet kábellel csatlakozik a készülékházhoz, így szabadon a lehető legkedvezőbb helyzetbe állítható. A ROSYTEXT klaviatúrán a teljes magyar betűkészlet megtalálható, és a numerikus billentyűk mellett speciális, szövegfeldolgozást segítő parancsbillentyűk is helyet kaptak. Ezek feliratozása jól átgondolt képi szimbolikán alapuló piktogrammák segítségével tükrözi az adott, általános parancsfunkciót.

Általában azt mondhatjuk, hogy az alapgép gondos konstruktőri munkát tükröz szinte minden részletében, és külső megjelenése is esztétikus.

A gép egykártyás központi egységében (ára 38 150 Ft) egy 4 MHz-es Z80 típusú mikroprocesszor, 64 kbájt RAM, 16 kbájt ROM, 1 db V24 típusú soros interfész és egy 8 bites párhuzamos interfész található. A nyomtatóhoz külön illesztőegység szükséges (10 510, ill. 19 230 Ft). A külön floppyillesztő egység maximálisan 4 db 8 hüvelykes meghajtót tud vezérelni (ára 22 490 Ft). A beépíthető floppymeghajtó mellett (ára 40 500 Ft) egy, két vagy három meghajtót tartalmazó külső bővítés vásárolható (ára rendre 97 950, 138 450 és 178 950 Ft), amelyet a gépasztal oldalára lehet szerelni.

Ajánlatos megvásárolni az opcionális lemezlekapcsoló egységek valamelyikét. A félautomatikus típus (ára 10 300 Ft) a lemez kihúzása után 30–40 másodperccel, az automatikus típus (ára 12 720 Ft) pedig a lemezhez fordulás után állítja le a meghajtót. A videomonitor ára 42 570 Ft, a billentyűzet 20 280 Ft-ba kerül. A részekre bontott árképzés több szempontból is ügyes fogás; az egyik szempont, hogy csak a szükséges minimumot kell megvásárolni. Egy nyomtató és nyomtatóillesztés, valamint ROSYTEXT programcsomag nélküli, minimális alapgép árát mi 174 520 forintra kalkuláltuk.

Az alapgép operációs rendszere a teljesen gyártóspecifikus, RODOS–I nevezetű lemezes operációs rendszer. Ez tulajdonképpen egy általános mikrogépes fejlesztő rendszer része. Ezt az is mutatja, hogy különböző mikroprocesszorokra kódot generálni képes assemblerek, illetve a BASIC mellett a PL/M és a Pascal nyelvek fordítóprogramjai is megvásárolhatók hozzá. Az operációs rendszert a ROSYTEXT felhasználója azonban gyakorlatilag nem látja, mivel a bekapcsolást közvetlenül követő rövid időtől eltekintve csak a ROSYTEXT programcsomaggal áll kapcsolatban (ára 38 470 Ft).

Formátumozás után a 8 hüvelykes floppyk szimpla írássűrűség esetén 220 000 karaktert, dupla írássűrűség esetén 450 000 karaktert képesek tárolni (a meghajtó és a vezérlő mindkét írássűrűséget tudja). Így 80 és 200 gépelt oldalnak megfelelő információ tárolható egy hajlékony mágneslemezen.

Nyomtatóként kétféle margarétakerekes és egy féllevelező minőségű mátrixnyomtatót szállít a cég. A robotron 6010-es írógép (ára illesztéssel együtt 38 000 Ft) 20 karakter/mp sebességű nyomtatást tesz lehetővé. Elsősorban ott ajánlható, ahol a napi nyomtatás mennyisége nem haladja meg a 100 oldalt.

A cég vállalja a már meglévő, robotron 6011-es szerkesztő írógépek illesztését is. Nagyobb teljesítményű nyomtatáshoz a 60 karakter/mp sebességű és robusztusabb konstrukciójú, robotron 1152-es nyomtatót szállítja a cég (ára illesztéstől függően 139 000, illetve 145 000 Ft). Mindkét margarétakerekes típusnak szép írásképe van, és 132 karakter maximális szélességben nyomtat. Kevésbé igényes, de rendszeres és nagy mennyiségű nyomtatáshoz ajánlható a 120 karakter/mp sebességű, TMT 120L típusú nyomtató (ára 58 000 Ft). Ennek előnye a közismert, egyszerű mátrixnyomtatókkal szemben, hogy egy külön üzemmóddan több menetben, az egyes menetenként kissé módosítva a tűsor pozícióját, nyomtatja az egyes sorokat. Így közel levél minőségű kiírást is lehetővé tesz.

Szövegfeldolgozás a memóriában

A ROSYTEXT programcsomag által megvalósított szövegfeldolgozás meghatározó jellegzetessége, hogy csak azokat a szövegeket tudja a felhasználó közvetlenül feldolgozni, amelyek a gép munkamemóriájában vannak. Ez azt jelenti, hogy maximálisan mindössze 9 sűrű oldalnyi (55 sor, soronként 60 karakter) szöveget lehet egy egységként kezelni. Hosszú szövegek kezelésének egyetlen módja, hogy a hajlékony mágneslemezen több fájlban helyezi el a felhasználó az egyes szövegrészeket. Ekkor azonban maga a szövegfeldolgozó semmiféle támogatást nem nyújt a felhasználónak ahhoz, hogy az ilyen részekből álló szövegek mind fizikai, mind logikai struktúrájukban egységes egészként kerülhessenek feldolgozásra. Nyomtatáskor például a felhasználónak kell ügyelnie arra, hogy az aktuális szövegrészt az utolsó fizikai lap végén elvágja, és átvigye a következő szövegrész (fájl) elejére. Mindez a ROSYTEXT egyik legkomolyabb korlátja.

Memóriában működve persze még viszonylag egyszerű algoritmusokat használva is elég hatékony feldolgozó szoftvert lehet készíteni. Így nem is túlságosan meglepő, hogy a ROSYTEXT villámgyorsan végzi el feladatait, ami előnyére szól. A szövegek egyszerű begépelésénél a sorokra tördelés automatikus, tehát nem kell a sorvégekre figyelni és „kocsi visszát" ütni, hanem a szöveg írása automatikusan folytatódik a következő sorban. A ROSYTEXT kiváló szolgáltatása, hogy biztosítja a sorvégi szavaknak a magyar helyesírás szabályainak megfelelő elválasztását. Ez – a cég szerint – az esetek 98 százalékában helyesen történik, csak az összetett szavaknál és az igekötős igéknél lehet probléma, amit a felhasználónak kell ellenőriznie és megfelelően korrigálnia.

Számlák és számtáblázatok kitöltéséhez igen hatékony támogatást nyújt a külön numerikus billentyűzet. Ha a „NUM LOCK" billentyűt lenyomja a felhasználó, akkor csak a numerikus billentyű (számjegyek és tizedespont), valamint a szóköz és tabulátor billentyűk hatásosak. Így tehát még véletlenül sem lehet olyan billentyűt használni, amely nem szükséges ehhez a speciális beviteli esethez. A szövegek bevitele során is lehet tabulálást alkalmazni. A tabulátor használata nagyon hasonló ahhoz, amit az írógépeken ismerünk, vagyis az előzőleg beállított tabulálási pontokra egy speciális tabulálási billentyű lenyomásával ugrunk.

A képernyős szövegfeldolgozásnak természetes előnye az a szabad formátummozgás, amit a fénymutató (cursor) tetszőleges helyre állításával érhetünk el. A megfelelő parancsbillentyűvel kombinálva a ROSYTEXT rendszernél kiválóan tudunk navigálni a szövegben. Ugorhatunk az aktuális sor elejére, végére, a képernyőn éppen látható szöveg első vagy utolsó sorának elejére. Ugrás hajtható végre a megelőző képernyőnyi szövegrészre vagy az aktuálisan látható szövegrészt követő, képernyőnyi szövegrészre. A képernyőn egyszerre 20 sor szöveg látható. Folyamatos bevitelnél a legalsó sor mindig kiszorítja a legrégebbi, legfelső sort. Néhány más, speciális ugrás mellett megadott számú oldal megadott sorának adott számú pozíciójára is lehet ugrani. Ezek az ugrások persze már inkább a meglévő szövegek karbantartása és módosítása során fontosak.

Egyszerű javítások felülírással, karakter vagy sor törlésével, netán karakter-közbeszúrással hajthatók végre. A javítás helyét a fénymutatóval kell kijelölnünk. A karakter-közbeszúrással vigyázni kell, mert ha a sor jobb oldali szélső karaktere eléri a sor végét a közbeszúrás miatt, akkor az újabb közbeszúrásoknál azok a karakterek, amelyek nem férnek el a sorban, egyszerűen elvesznek. Az igen jól szerkesztett kézikönyv erre teljesen korrekten felhívja a figyelmet, bár itt egyértelműen koncepcionális hibáról van szó, aminek nem lenne szabad előfordulnia a termékben. Az egyszerű javításokhoz tartozik még egy sor közbeszúrása vagy a memóriában lévő szöveg törlése.

Magasabb szintű funkciók

Az ún. címszerkesztési funkcióval lehet elérni, hogy a begépelt sorok centrikusan helyezkedjenek el. A szedőszerkesztés nevű funkció üzemmódjában végezhető el a begépelt szöveg sorkiegyenlítése, azaz a sorvégek „kiegyenesítése". Ekkor a ROSYTEXT megfelelően arányos elhelyezésben szóközöket tesz az egyes sorokba. A szedőszerkesztés speciális lehetősége, hogy soronként is dönthet a felhasználó a kiegyenlítés szükségességéről.

A rendező szerkesztésnek nevezett funkció a ROSYTEXT meglehetősen egyedülálló sajátossága. Véleményünk szerint ugyanakkor nem valami rendkívüli szolgáltatásról, hanem a kifogásolt koncepcionális termékhibák korrukciós korrigálásáról van szó. Olyan jelentős változtatások végrehajtására ajánlja ezt a gyártó, amelyek – megfogalmazása szerint – megváltoztatják a szöveg formátumát. Példaként a sorból a közbeszúrások miatt kicsorgó karakterek esetét említi. A rendező szerkesztési üzemmód ezen úgy segít, hogy az üzemmódba belépve két üres sor iktatódik be a fénymutatóval kijelölt sor mögé, és a fénymutató pozíciója utáni további sorrész lesz a két üres sor utáni első sor (abban a sorban viszont a fénymutató előtti karakterek lesznek üresek). Így teremt üres helyet a nagyobb beszúrásokhoz a ROSYTEXT. Miután a felhasználó elvégezte a beszúrást, egy speciális mozgató billentyűvel lehet összezárni ismét a szöveget. A második szövegrész karaktersorozata, mint valami kígyó, szemmel is követhető ütemben zárkózik fel az első szövegrészhez. Ez igen látványos, de konstrukciós szempontból meglehetősen értelmetlen. Ráadásul eközben a sorvégi elválasztó jelek és a sorkiegyenlítést szolgáló szóközök elnyelődnek, így a formátumot ezt követően ismételten csinosítani kell.

Egy sor más, magasabb szintű funkció is van. Ezeket már csak címszavakban említjük: aláhúzás, kiemelés (a képernyőn erősebb fény, a nyomtatón dupla nyomtatás), aláhúzás és kiemelés, sorok mozgatása, keresés és pozicionálás megadott karaktersorozat szerint, lemezkatalógus-behívás, szövegkivitel, szövegbevitel, szöveg törlése lemezen, szövegrész kivitele lemezre stb. A nyomtatási lehetőségek gyakorlatilag

minden igényt kielégítenek, és a laponkénti nyomtatástól kezdve a több példányos és számozott lapos nyomtatásig terjednek. Itt említjük meg, hogy a ROSYTEXT a nyomtatás során végzi az igazi lapokra tördelést. A képernyőn csak az aktuális oldalszám alapján tudhatjuk, hogy melyik lapon vagyunk.

A rendszer külön szolgáltatása az ún. törzsszövegek kezelése. Gyakran ismétlődő szövegeket lehet ily módon többszörösen felhasználhatóvá tenni. A törzsszövegek, illetve törzsszöveg-állományok egy különálló munkamemóriában helyezkednek el, és ugyanúgy lehet feldolgozni (bevinni a lemezről, szerkeszteni stb.) azokat, mint a normál szövegeket. Billentyűzetről beadandó és rögtön végrehajtott parancsokkal vagy speciális címkézéssel azonosított törzsszövegrészeket vagy törzsszövegsorokat lehet egyenként átvinni a szöveg megfelelő helyére.

Ez a szolgáltatás csak a legelemibb törzsszöveghasználatot támogatja. Nem lehet hosszú címlistákkal vagy névjegyzékekkel feldolgozást végezni, ráadásul az ismétlődő törzsszöveg-feldolgozás nem fogalmazható meg úgy, hogy a rutinszerű dolgok automatikus láncolással menjenek végbe.

Összefoglalva:

A ROSYTEXT rendszer kiváló vállalati munka eredménye, és kiváló termék lenne, ha nem lennének számomra bosszantó hibái. Eddig még nem fogalmaztam meg kritikai véleményt ennyire keményen a Termékismertető rovatban. Most is csak azért teszem, mert a piaci munka, a gyártási műgond, az oktatás, szerviz és vevőszolgálat tekintetében annyival a többiek fölé emelkedik a ROLITRON, hogy egyszerűen nem lehet elhallgatni a szoftver elvi (és nem működésbeli) gyengeségeit. Azért sem szabad erről hallgatni, mert operációs rendszere gyártóspecifikus, tehát a vevőnek még választási alternatívája sincs sem a szövegfeldolgozó, sem más szoftver vonatkozásában.

Az elvi lehetőségekhez képest korlátozott felhasználásban (kisebb volumenű szövegfeldolgozás, nincsenek hosszú és/vagy rutinszerűen ismétlődő szövegek) mégis kiválóan lehet alkalmazni. Két véleményünk között nincs ellentmondás. Az utóbbi esetben ugyanis kis volumenű és kis teljesítményigényű feldolgozásokról van szó, ahol a képernyős szövegfeldolgozók világában már elért teljesítőképességhez képest korlátozott ROSYTEXT konstrukció is bőven megfelel.

Különösen igaz ez azért, mert a rendszer képernyőinterfésze kiválóan kivitelezett és könnyen elsajátítható. A legfelső, ún. tabulátor sorban a beállított tabulátorhelyek jól láthatók. A következő, ún. információ sor mutatja a szöveg nevét, a fénymutató helyét (oldal-sor-pozíció alakban), valamint a szövegformátum paramétereit. Így a két felső sor alapján mindig mindent tudunk, ami az ezt követő, 20 soros szövegterületre vonatkozik. Az alsó két sor közül a felsőben a leütött parancsbillentyűk teljes, magyar nyelvű parancsszavait és az azokhoz megadott paramétereket látjuk. A legalsó sor az ún. visszhangsor, amelyben az utoljára leütött 20 billentyűt látjuk megjelenítve. Így ha munkánkat kénytelenek vagyunk megszakítani, akkor is mindig tudni fogjuk, hogy milyen állapotban hagytuk abba.

Egyértelmű tehát, hogy a lehetséges alkalmazások egy részénél a ROSYTEXT kiválóan használható termék. Senki se gondolja azonban, hogy ez a rendszer a csúcs. Még személyi számítógépre kidolgozott szövegfeldolgozók között is egy sor olyan terméket lehet találni, amely ugyanilyen hardveradottságok mellett magasan túlszárnyalja a ROSYTEXT lehetőségeit. Érdemes tehát vásárlás előtt alaposan tájékozódni, és csak azután dönteni a vételről.

Reméljük, hogy a ROLITRON is több pénzt, időt és energiát fog fordítani egy nagy teljesítményű ROSYTEXT–II – szerintünk szükséges – kifejlesztésére. Titokban abban is bízunk, hogy lemond gyártóspecifikus alapszoftveréről, és áttér a CP/M-re.

NACSA SÁNDOR

Szövegfeldolgozók – objektum-orientáltan

A számítógépes szövegfeldolgozók konstrukciós fejlettsége alapvetően múlik azon, hogy tervezői milyen mértékben vették észre és milyen mértékben tudták érvényesíteni az alkalmazásban rejlő természetes struktúráltságot. Szoftverkonstrukcióknál azonban mindig is túl nagy a csábítás, hogy az egyszerűbben megfogható, *művelet-orientált* szemléletben határozzák meg a különböző szolgáltatásokat. A szoftverkonstruálás egyik legkorszerűbb, *objektum-orientált* módja ráadásul még kutató-fejlesztő berkekben is alig ismert megközelítés.

Az objektum-orientáltság a strukturált gondolkodás egyik legmagasabb szintje. A struktúrát alkotó elemek és egymáshoz képesti logikai viszonyaik, valamint a hordozón való elhelyezkedésük és fizikai viszonyaik sokkal átfogóbb rendszert alkotnak, mint önmagukban az egyes műveletek rendszere. Az előbbiben a műveletek és az általuk meghatározott viselkedés a teljes objektumrendszer szerves, elválaszthatatlan részét képezik. Az utóbbi esetben az objektumok elvesznek a különböző műveletek, elsősorban helyileg átgondolt, gyakran az általános használati logikát is nélkülöző, meglehetősen szerteágazó rendszerében. Végeredményben a paraméterek szintjére degradálódik mindaz, ami a leginkább meghatározó és a leglényegesebb.

Szövegfeldolgozók hatékony megvalósítását, használhatóságát és elsajátíthatóságát alapvetően meghatározza a szövegnek, mint egy összetett struktúrájú és viselkedésű objektumnak a konstrukció központjába való helyezése. Az írott, nyomtatott és képernyőn megjelenített módon megformált, egységes nyelvi mondanivaló különböző részekből áll, amelyek tartalmukban egy logikai egységet, megjelenésükben pedig egy fizikai egységet alkotnak. Az írásbeliség hosszú időszaka alatt kialakultak azok az *egységes logikai és fizikai formák*, amelyek egymással igen magas fokú megfelelésben állva, lehetővé teszik a mondanivaló átgondolt kifejtését és karbantartását (módosítását, átrendezését stb.), valamint annak széles körű és hatékony átvételét.

A képernyős szövegfeldolgozók az általánosan elfogadott logikai és fizikai formák kezelésének és feldolgozásának jól automatizálható részfeladatait gépesítik. Történik mindez nem egy elvont hatású következményekkel járó parancsnyelvi szinten, hanem a közvetlen manipuláció és a vizuális visszacsatolás természetes gépi környezetében. Egy képernyős szövegfeldolgozó korszerűsége elsősorban azon múlik, hogy konstruktőrei mennyire tudták felfogni és érvényesíteni ezt a két alapgondolatot. A megszokottól való elrugaszkodás nehézségét jól mutatják a „képernyősített" írógépből és a hagyományos, parancsnyelv-orientált szövegszerkesztőkből kiinduló konstrukciók.

Az egységes logikai formákat mindenki ismeri. Egy szövegnek *szakaszai* vannak, amelyek egy önmagában is megálló tartalmi egységet jelentenek. A nagyobb szakaszokat, amelyeket a hatékonyabb azonosítás érdekében külön címmel (fejrésszel) és esetleg számmal látunk el, *fejezeteknek* szoktuk nevezni. Vannak olyan speciális szakaszok, amelyek az egész szöveg szempontjából egy-egy célfunkciót látnak el. Ilyenek a *bevezető*, az *előszó*, a *mellékletek*, *tartalomjegyzék*, *névmutató*, *tárgymutató*, *irodalomjegyzék* stb. Van ezek között olyan, amelyikből több is lehet egy szövegen belül, a szövegek többségéből azonban java részük hiányzik. Rövid szövegeknél (levél, emlékeztető, feljegyzés stb.) gyakran előfordul, hogy nincsenek fejezetek, csak számozott szakaszok, és a megfogalmazók valahol elől, a folyamatos szövegrészek valamelyikéből hivatkoznak ezekre.

A szöveg szakaszokra darabolása általában úgy történik, hogy a szakaszokon (ill. fejezeteken) belül kisebb szakaszok vannak, amelyek részét képezik a nagyobb szakasz (fejezet) tartalmi egységének. A tartalmi egységek beágyazott struktúrájának legalsó szintjén található szakaszok egy vagy több *bekezdésből* állnak. Ez a kifejtett gondolatok legalsó szintű csoportját jelenti. A bekezdések *mondatokból* állnak, a mondatok *szavakból*, a szavak pedig *írásjelekből*.

Egyébként azt is lehetett volna mondani, hogy a bekezdés a szakasz egy speciális típusa, ugyanúgy, mint ahogy a fejezet is egy (másféle) speciális típusa a szakasznak. Azért is említjük ezt, mert kiválóan példázza, hogy az *objektumok tiszta fogalmi rendszere* szempontjából milyen jellegű szemléletmódra van szükség. A buta automatának tekinthető szövegfeldolgozó a mindennapi használat logikája szempontjából akár igen komoly ellentmondásokat is tartalmazhat, ha ez nem érvényesül benne. Az általánosan elfogadott fizikai formák sem juthatnak kellően érvényre, ha a szövegfeldolgozó által ismert logikai formák formailag nem elég fejlett és nem elég tiszta rendszert alkotnak.

A fizikai formák teljesen közismertek. A szöveg *oldalak* egymásutánján helyezkedik el, az oldalakon belül pedig *sorok* egymásutánjaként. A sorok *írásjelek* egymásutánját jelentik. A logikai és fizikai formák rendszere az írásjelek szintjén találkozik tehát egymással. Az ilyen megfelelésnek vannak bonyolultabb esetei is. Az ún. fő fejezetek más fő fejezetekkel nem átfedésben lévő oldalakon találhatók. Itt is megemlítjük, hogy a fizikai formák kialakult fogalmi rendszere ennél sokkal bonyolultabb is lehet (lásd például a µM több hasábos oldalstruktúráját).

A logikai és fizikai objektumok rendszerének fogalmi meghatározásához és ismeretéhez szorosan hozzá tartozik az egyes típusok tulajdonságjellemzőinek ismerete. Ilyen például egy oldal sorainak száma, illetve a sorok közötti távolság. Egy logikai objektum fizikai jellemzőjére jó példa az, hogy mennyivel beljebb kezdődik egy bekezdés első sora. Az írásjel esetében annak típusa és nagy, illetve kisbetűssége a főbb jellemzők. Az összetett objektumoknál viszont a legfontosabb jellemző maga a struktúra, ami az objektumot alkothatja. Az egyes objektumtípusok általános strukturális modelljét és a típusokhoz tartozó elemi jellemzőket bárki feltárhatja magának, csak a különböző géppel írott vagy nyomtatott szövegmintákat kell tanulmányoznia (akár a µM-t).

Egy szövegfeldolgozó megismeréséhez is először az kell, hogy megállapítsuk, milyen logikai és fizikai formákat ismer. Ezek után azt kell megnéznünk, hogy e formák kezeléséből mit automatizál. Ennek néhány tipikus esete: automatikus sorokra és lapokra tördelés (nem kell tehát ügyelnünk ezekre a határokra, például kocsi visszát ütni), automatikus elválasztás, automatikus lábjegyzetelhelyezés és számozás stb. Igen sok kezeléshez kap hatékony manipulatív segítséget a felhasználó: kiemelések, fejléckészítés, kézi laptördelés, szövegrészek módosítása és áthelyezése, különböző objektumok elemi jellemzőinek meghatározása stb. Konkrét termékekről az 1984/4. szám 3. oldalán, az 1985/1. szám 26. oldalán és az 1985/2. szám 8. oldalán olvashattak korábban a µM-ban. Az 1985/2. szám 24. oldalán a dokumentumfeldolgozás részeként említettük.

Az objektum-orientált szemlélettel tehát nemcsak konstruálni lehet szövegfeldolgozót, hanem ezzel a módszerrel mindenki tudatosan fel tudja mérni saját igényeit, értékelni tudja a kínált megoldásokat, és ki tudja választani a neki legmegfelelőbbet.

Az objektum-orientáltsággal az előző szám 25. oldalán szereplő téglafalrajzoló programnál is találkozhattak. Kérdésünk, hogy mi az ott közölt programszöveg logikai és fizikai struktúrája?

NINO

Alapozás XII.

Bebizonyítottuk, hogy bizonyos egyszerű függvények függvényértékeinek kiszámítását elvégző operátorcsapatokat képesek vagyunk létrehozni, és ezeknek az operátorcsapatoknak megfelelő kapcsolások megszerkesztésére is adtunk módszert. A digitális gép megépítésében nélkülözhetetlen szerepű építőelemek révén rendelkezésünkre álló fontos lehetőségek pontos elméleti és gyakorlati megismerése lesz a következő feladatunk.

Egy egyszerű, de nagyon hasznos módszer

Mielőtt a problémakör nehezebb részével, az operátoroknak (és operátorcsapatoknak) a gyakorlat igényeit kielégítő pontosságú matematikai modelljeivel foglalkoznánk, egy nagyon hasznos módszert ajánlunk a megismert pneumatikus kapcsolások „életjelenségeivel" való alaposabb megismerkedés érdekében.

A módszer lényege: megrajzolni a levegőrészecskék áramlási pályáját. (Ez természetesen csak vázlatosan sikerülhet, de a pontos munka most nem is szükséges, elegendő a nagyvonalú vázlat is.)

Az *1. ábrán* a két operátor egyikének egyik bemeneti fúvókáján keresztül sincs légáramoltatás, a másik operátor egyik bemeneti fúvókáján keresztül viszont van. A kialakuló légáramlási pályákat – egyszerűsítetten – vastag vonal jelzi.

A *2. ábra* és a *3. ábra* két, kissé összetettebb operátorcsapatban kialakuló légáramlások pályáit mutatja.

Az olvasó feladata addig szerkesztgetni operátorcsapatokat, és minden lehetséges bemeneti „fúj"–„nem fúj" változat esetében megrajzolni a légáramlások pályáját, amíg már puszta ránézésre is meg tudja mondani, hogy mikor mely kimeneti vezetékekben van „huzat", illetve mely másokban nincs.

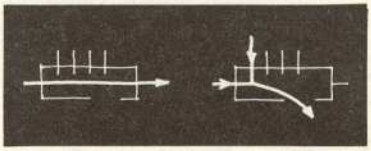
Nem lesz haszontalan ez a munka, abból a szempontból sem, hogy olyan kapcsolásokra is rá lehet bukkanni, amelyeknek működését nem lehet egy szempillantás alatt átlátni. (Hogyan működik például a *4. ábra* két nagyon egyszerű rendszere?)

Miután kellő jártasságot és gyors eligazodóképességet szereztünk egyszerű kapcsolások áttekintésében, rátérhetünk a működés finomabb elemzésére. Kezdjük az „ideális", a végtelenül fürge és pontos matematikai modell vizsgálatával.

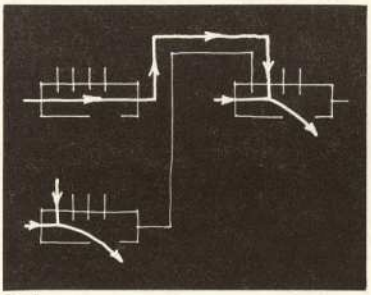
Az „ideális" modell

Ezt a modellt azért nevezzük „ideálisnak", mert eszerint pneumatikus operátorunk a bemenetén zajló folyamatokra késés nélkül, azonnal, végtelenül fürgén reagál. (Az ideális szót a tudomány nagyon sokféle értelemben használja. Majdnem olyan sokféle értelemben, mint az utcanyelv az „izé"-t. Ez természetesen a fogalom általános értékességéből, általános hasznosságából sokat levon, hiszen kinek mikor mi az ideális – az lényegében csak őtőle függ. Sokszor még nem is definiálják pontosan, hogy mit értenek e szón.)

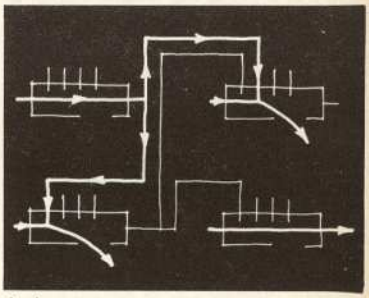
Hozzávesszük még modellünk „ideális"-ságához azt is, hogy a kapcsolás minden pontja késedelem nélkül azonnal tudjon, ha kell, egyik állapotból a másikba váltani. Látható, hogy az „ideális" szó most a lomhaságnélküliséget, a végtelenül fürgeséget kívánja jelenteni. (Első pillanatra úgy tűnik, hogy nagyon jó volna, ha volna ilyen. Látni fogjuk azonban, hogy a tehetetlenség, a lomhaság létfontosságú tulajdonság. A valóban „ideális" az volna, ha rendelkezésünkre állna bármilyen kicsi tehetetlenségű, bármilyen kicsi lomhaságú, de tehetetlen, lomha rendszer, vagy ezeket a tulajdonságokat tetszőlegesen tudnánk alakítani, mindig a kívánt lomhaságú rendszert előállítani. Ezek a lehetőségek azonban, sajnos, csak az „ideák", helyesebben az ábrándok világában állnak rendelkezésre.)



1. ábra

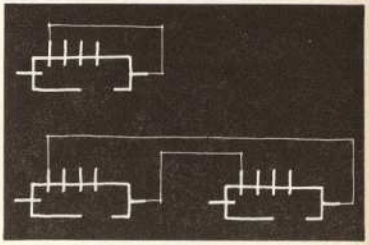


2. ábra

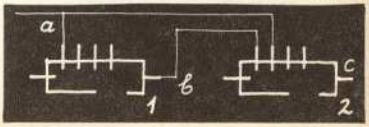


3. ábra

4. ábra



5. ábra



Ezek után rátérünk pneumatikus operátorunk viselkedésének tanulmányozására, feltételezve, hogy működését „az ideális modell" pontosan írja le.

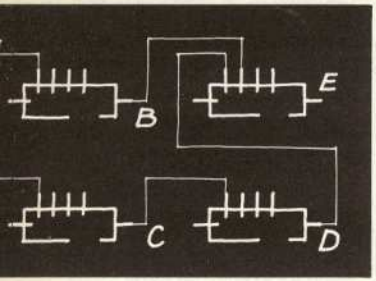
Pneumatikus operátorunk működése során két tevékenység biztosan állandóan zajlik,

- a bemeneti pontok állandó és szakadatlan figyelemmel kísérése, és
- az állandó és szakadatlan gondoskodás a megfelelő kimeneti értékekről (állapotokról).

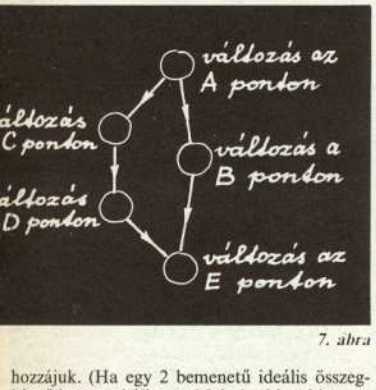
Az „ideális", azonnali reakciót feltételező modell szerint, az operátor kimenetén bemeneti változás nélkül nem történhet változás. (Bemeneti, illetve kimeneti változáson a bemeneti, illetve kimeneti érték, illetve állapot megváltozásának jelenségét értjük.) Mivel a kimeneti változás szükséges feltétele a bemeneti változás, más szóval bemeneti változás nélkül nincs kimeneti változás, ezért az operátor csak bemeneti változáskor kerülhet abba a helyzetbe, hogy kénytelen kimeneteinek állapotán változtatni.

Az ideális modell szerint az operátor figyelemmel kíséri bemeneteinek állapotát; ha ezek nem változnak, az operátor sem változtatja kimeneteinek állapotát. Ha a bemeneti állapot megváltozik, lehet (de nem biztos), hogy igazítani kell a kimeneti állapoton is. Azaz lehet, hogy korrigálni kell, hogy hozzá kell igazítani a kimeneti állapotot az új, a megváltozott (változás utáni) bemeneti állapothoz. Ilyen igazításra, korrigálásra azonban – ismételjük – csak bemeneti változáskor kényszerülhet az operátor, máskor nem.

Az igazítási, korrigálási kényszer viszont akkor biztosan fellép, ha az operátornak a változás előtti és a változás utáni bemeneti állapota olyan, hogy más-más kimeneti állapot tartozik



6. ábra



7. ábra

hozzájuk. (Ha egy 2 bemenetű ideális összegképző bemeneti állapotai 0 és 1, akkor kimenetének állapota 1. Ha a bemeneti állapot (0;1)-ről (1;0)-ra változik, a kimeneti állapotnak nyilvánvalóan nem kell megváltoznia. Ha viszont a bemeneti állapot (0;1)-ről nem (1;0)-ra, hanem például (1;1)-re változik, az operátor biztosan kimenetiérték-korrekcióra, kimenetiérték-változtatásra kényszerül.)

A módosítási, a korrekciós kényszer nyilvánvalóan az állapotváltozásokkal halad együtt (lépeget, terjed), azokat sem meg nem előzheti, sem le nem maradhat tőlük. (Ez utóbbi a modell „ideális" volta miatt van így.)

A korrekciós kényszerről a bemenetei valamelyikének megváltozása révén szerezhet és szerez tudomást az operátor.

Az *5. ábrán* a *b* bemeneti pont értéke 0-ról 1-re változik. Nézzük meg, hogy e változás az operátorokat hogyan hozza kimenet-korrekciós kényszerbe! A kényszerbehozás rendje nyilvánvalóan azzal függ össze, hogy mely operátor mikor (mi után) szerezhet tudomást valamely bemenetén létrejött állapotváltozásról. Mivel az állapotváltozás tudomásulvétele egybeesik az állapotváltozással, azaz azt késés nélkül követi, az állapotváltozások rendjére kell figyelnünk, ha arra vagyunk kíváncsiak, hogy mikor mely operátor miről szerez tudomást.

Mivel a tudomásulvételt késés nélkül követi a kimeneti érték kialakítása (korrigálása, igazítása) és a kimeneti érték is állapot, a kimenetiérték-kialakítás rendje beletartozik az állapotmegváltozások rendjébe. Ha minden pontot valamilyen operátor kimenetének fogunk fel, e kimeneteket pedig indikátorváltozóknak, azaz olyan jellemzőknek, amelyek a bemeneti jellemzők együttesét jellemzik valamilyen módon, akkor – remélhetjük, hogy – az előbbi folyamatok révén arra is választ kapunk, hogy mikor mely pont mit indikál egy állapotváltozás és annak következményei létrejötte (kialakulása) folyamán.

Az 5. ábrán a bemeneti pont értéke (állapota) tehát egyről nullára változik. Vizsgáljuk meg az állapotkialakulásokat aszerint, hogy mikor melyik operátor kerül kimenetiérték-felülvizsgálati, illetve -módosítási kényszer állapotába!

Összpontosítsuk figyelmünket a jelenségekre, a történésekre, illetve az ezek közötti kapcsolatokra! A folyamatot a következő eseménynaplóval írhatjuk le.

A változás előtti állapot:
a = 1, b = 0, c = 0.

Az állapotváltozást közvetlenül követő állapot:
a = 0,
azaz megváltozott két operátor bemeneti állapota. Mindkettő kimenetiérték-korrekciós kényszerbe kerül. A kérdés most az, hogy az operátorok a korrekciót milyen bemeneti értékek alapján végzik el. A 2. operátor első bemenetén eddig 0, a másodikon pedig 1 volt, most a második bemeneti érték 0-vá vált. E pillanatban tehát a kimenet értéke 1. Azaz a = 0, c = 1.

Ezt az állapotot azonban nyilvánvalóan röstön „elfújja" az 1. operátor kimeneti értékének kialakulása, aminek következtében az
a = 0, b = 1, c = 0
állapot alakul ki.

Azt kell tehát mondanunk, hogy 0 időtartam alatt *c* értéke 0-ból 1-be, majd újból 0-ba alakul. Előfordulhat-e ez a valóságban? Milyen hatásai lehetnek ennek a jelenségnek? Lehet-e egyáltalán hatása valamire? Ha igen, hogyan igazolhatnánk létét kísérletileg? (Ha nincs hatása, nyilván igazolni sem tudhatjuk.) Egy pillanatnyi változásból kettő, szintén pillanatnyi változást csinál a rendszer.

Ha valakinek ezek az okoskodások gyanúsak, furcsáknak tűnnek, annak lehet még furcsábbakat is mutatni. (Egy pillanatnyi változásból például szintén egy pillanat alatt lejátszódó tetszőleges sok változás előállítása sem tűnik nehéznek.)

Ha valaki erre az elemzésre azt mondaná, hogy miért nem várja meg a 2. operátor, amíg az 1. operátor kialakítja az „új" kimeneti értékét, akkor több kérdést is feltehetünk az illetőnek. Honnan kellene tudnia a 2. operátornak, hogy „meddig" kell várnia? A másik pedig: honnan kellene tudnia a 2. operátornak, hogy a másik (azaz 1-ből jövő) bemenetén zajló „folyamatnál" meddig kell várnia? És a harmadik: a várakozás alatt mi fogja definiálni az operátor kimeneti értékét?

Vegyük észre, hogy az összes kérdés jogos. És azt is, hogy egyikre sem tudunk válaszolni. Ki kell mondanunk, hogy *az „ideális" modell önmagában nem elegendő minden kérdés megnyugtató elintézésére, de még kizárására sem.*

A *6. ábra* példája még jobban megerősíti az előbbiek során kialakult bizonytalanságérzetünket.

Az eseménynaplóval birkózzon egy kicsit az olvasó! Megéri. Mi itt csak az operátorok korrekciós kényszerbe kerülésének rendjét ábrázoltuk *(7. ábra)*.

A 7. ábra az események, a történések rendjét szemlélteti. Ha egy esemény jeléből (kör) irányított szakasz indul ki, akkor ez az esemény (történés) korrekciós kényszerbe hozza azt az operátort, amelynek kimenete megváltozása (esetleg csak felülvizsgálata) eseményének jelébe mutat a nyíl (fut be a vonalszakasz).

Az ábráról leolvashatunk olyan információkat, hogy melyik esemény melyik másikat előzi meg. Rögtön látszik azonban a modell már említett hiányából eredő elégtelenség is, az, hogy nem minden esetben kapunk eligazítást a megelőzési (előzményi) viszonyokról. Különösen érezzük ennek hiányát az ábra legalsó eseményével (történésével) kapcsolatban. Három eset fordulhat elő ugyanis.

Az egyik ágról jövő korrekciós kényszer megelőzi a másikat, a másik ágról jövő korrekciós kényszer megelőzi az egyiket, a két korrekciós kényszer egyszerre érvényesül. Az első két esetben az 5. ábra rendszere bemeneti változásának megfelelő kimeneti érték kialakítása, előfordulhat, hogy 2 egymás utáni történés formájában valósul meg.

Felvetődik a kérdés, hogy miért tartjuk ezt a bizonytalanságot olyan veszedelmesnek. A válasz egyszerű. Ha a rendszerünk kimeneti folyamatait figyelő rendszer szempontjából nézzük a kérdést, nem mindegy, hogy egy ilyen rendszer milyen bemeneti információkat kap.

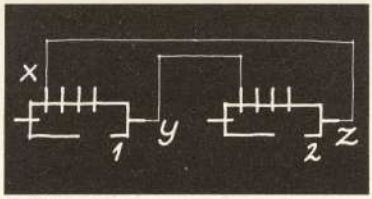
Az „ideális" modellről tehát megállapíthatjuk, hogy az általa szolgáltatott információk nem elegendők a rendszer teljes viselkedésének leírására.

Tehetünk erőfeszítéseket a modell toldozásfoldozása érdekében, olyan új megállapodások bevezetésével, amelyeknek segítségével a többértelműségek megszüntethetők. Ez az egyébként érdekes út sem vezet azonban túl messze. A következő nagyon egyszerű példák mutatják, hogy a végtelenül fürge működés önmagában is forrása bizonyos ellentmondásoknak, tehát maguk a végtelenül gyors reakciók lesznek kerülendők.

A *8. ábra* az ideális modell szerint egy egyenlet rajzi leírása.

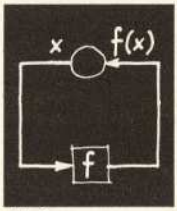
Az ideális modell egy matematikai alakja, mint ismerjük,
$k = 1 - \max(b_1, b_2, b_3, b_4)$.
A 8. ábrában b_2, b_3 és b_4 zérus, tehát a maximum értéke b_1 értékével egyezik meg. Mivel b_1 és k értéke – összekötöttségük miatt – egyenlő, a
$k = 1 - k$
egyenletet kapjuk. Ezt az egyenletet pedig k-nak (illetve b_1-nek) sem a nulla, sem az 1 értéke nem elégíti ki. Modellünk szerint ez a kapcsolás nem fog digitális rendszerként működni. A modell szerint stabil állapotba kerül a rendszer, a kimenetén – és bemenetén is – félerővel fog „fújni" a vezetékekben áramló szél. Ennek a fizikai tapasztalással való igazolása azonban közel sem tartozik az egyszerűen elintézhető kérdések közé. Az azonban biztos, hogy vagy kivezet a digitális körből a 8. ábra rendkívül egyszerű kapcsolása, vagy ha nem, a modell hibásan írja le a rendszer működését.

A 9. *ábrából* másfajta, de ugyancsak meglepő következtetésekre juthatunk. A 9. ábra formulás egyenletét a következőképpen kaphatjuk. Az 1. operátor egyenlete $y = 1 - x$, a másodiké pedig $z = 1 - y$. Ezekhez hozzá kell még venni a $z = x$ összefüggést is, mert a 2. operátor kimenete megegyezik az 1. operátor bemenetével. E három összefüggés azonban – az összes ami rendelkezésünkre áll – nem határozza meg egyértelműen a rendszer működését.



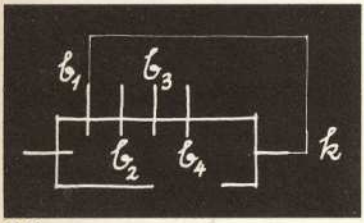
9. ábra

Természetesen az egyenletrendszer megoldását (megoldásait) folyamatként kell értelmeznünk. Így jó megoldás x-re például az állandóan 0 és az állandóan 1 folyamat is, sőt az összes olyan folyamat is, amelynek értékei 0 vagy 1 lehetnek. Nyilvánvalóan kielégítik a 9. ábra egyenletét az olyan oszcilláló folyamatok, amelyek értéke hol 0, hol 1, tetszőleges módon



10. ábra

11. ábra



8. ábra

váltakozva. Az ilyesmi pedig legalábbis furcsa egy fizikai rendszer életében, ahol – hitünk szerint – mindennek megvan az elegendő oka. (Megjegyezzük, hogy maradi gyanakvásunkat az is tovább táplálja, hogy ilyesféle jelenségeket az anyagi világban eddig sohasem tapasztaltunk.)

Úgy véljük, ennyi teljesen elég ahhoz, hogy hagyjuk az „ideális" modellt, és keressünk helyette reálisabbat. Megemlítünk azonban még egy okot, ami ezt az elhatározásunkat erősíti. A digitális gép információkkal dolgozik. Így minden operátor kimenete indikál valamit, informál valamiről. Mihez kezdhetnénk az olyan rendszerekkel, amelyeknek „életében" előfordulhatnak 0 és 1 értékekből teljesen önkényesen összeálló oszcilláló folyamatok is, mint amilyenekről már említést tettünk. Miről informálna egy ilyen oszcilláló értékű jellemző? (A probléma azonban szerencsére nem az, hogy ilyen rendszerekkel kellene valamit csinálnunk, mert azok csak modelljük szerint – a valóságban nem – ilyen, semmire (?) sem használható viselkedésűek.

Egy megjegyzés az implicit függvényekről

A 8. ábra és a 9. ábra közös jellemzője az, hogy egy rendszer egy kimeneti jellemzője (pontja) azonos egy bemeneti jellemzőjével (pontjával). (Esetünkben mindegyik rendszernek egyetlen bemeneti és egyetlen kimeneti pontja van.) Ha a rendszer által megvalósított transzformáció olyan, hogy az *x*-szel jelölt bemeneti értékből az $f(x)$-szel jelölt kimeneti értéket hozza létre, akkor, mivel a kimeneti érték azonos a bemenetivel, a következő összefüggést (egyenletet) kapjuk a rendszer bemeneti és kimeneti értékére *(10. ábra)* $x = f(x)$.

Az ilyen egyenleteket (kapcsolatokat) implicit egyenleteknek (kapcsolatoknak) hívják. Ritkán esik róluk szó. Az ok: noha a természetben rengetegszer előfordulnak, de mivel megoldásuk nehéz szokott lenni, nem szoktak róluk tudomást venni. Ritka az olyan ember, aki szívesen emleget olyasmit, amivel alig vagy sehogyan sem boldogul. Az implicit összefüggések között pedig gyakori az olyan, amely nem gondoskodik kellő sikerélményről a vele foglalkozók számára.

A *11. ábra* rendkívül egyszerű rendszere, illetve kapcsolása, melyben egy négyzetreemelő kimenete azonos bemenetével, az $x = x^2$ egyenlet rajzi megfelelője. Megoldás lehet az azonosan 0 folyamat, az azonosan 1 folyamat, vagy az összes, a 0 és 1 között bárhogyan oszcilláló folyamat is. Ilyenre már láttunk példát.

Ha fizikai rendszerről van szó, és *x* ennek egy állapotjellemzője, nem csoda, hogy gyanúval fogadjuk az ilyen megoldásokat, hiszen a fizikai rendszerek között – hitünk szerint – gyakori szokás, hogy egyszerre csak egy állapotban vannak.

Gondoljuk végig, mire mennénk egy olyan rendszerrel, amelyben néhány ilyen rész működne. Kerülésük tehát általában jogos, agyonhallgatásuk azonban pusztán hiúság és tudáshiány.

Következő lépésünk az lesz majd, hogy kilábalva a túlzott egyszerűsítésekből származó zavaros helyzetből, olyan modelleket szerkesztünk és vizsgálunk, amelyek nem végtelenül fürge működésűek. Látni fogjuk, hogy ez az út járható lesz, és egyszerű, könnyen kezelhető modellek szerkesztésére nyújt majd lehetőséget.

POGÁNY CSABA

Móricz Sándor, az MTA SZTAKI Cosy leányvállalatának 38 éves igazgatója, gépészmérnök–közgazdászmérnök végzettségű. Első munkahelyén, a Gépkísérleti Intézetben kezdett a számítástechnika gyakorlati részével foglalkozni. Később részt vett a SZÁMOK első rendszerszervezési tanfolyamán, és ezen a területen tevékenykedett a Ganz-MÁVAG-nál. Ezt követően a SZOT Társadalombiztosítási Igazgatóság hívására az első R20-as gépek üzembe állításával foglalkozott, mint hardver vezető, majd az akkor, hét nagy államigazgatási tárca összefogásával beinduló Államigazgatási Számítógépes Szolgálatnál a SZOT képviselője lett technikai vonalon. Irányítóként vett részt az AFIT számítástechnikai bázisának és vállalati rendszerének létrehozásában, megszervezésében. Az AFIT megszűnése után a Microkey mérnöki irodájának vezetője lett. A Microkey kutatási-fejlesztési társulás az MTA SZTAKI-nál folyó fejlesztést, innovációt bonyolítja, ideértve a gyártás és a forgalmazás megszervezését is. Tagjai a SZTAKI-n kívül az Elektromodul és a Sárisápi TSZ. Mikor a SZTAKI menedzser leányvállalata, a Cosy megalakult, Móricz Sándort hívták meg igazgatónak, és jelenleg ebben a beosztásban dolgozik.

– Alig másfél évtizedes pályafutása alatt a számítástechnika jó néhány területén dolgozott, elég változatos beosztásokban. Gondolom, hogy ez bizonyos rálátást ad a szakma egészére, problémáira.

– Úgy van. Bár mindig rengeteget tanultam, legtöbbet mégis abból profitáltam, amit az élet sűrűjében, legtöbbször a saját bőrömön tapasztaltam. Soha nem kötött le, hogy leüljek egy irodában és elméleti szinten próbáljak megoldani egy problémát. Be kell menni a „vadak közé", így lehet a legtöbbet nyerni, bár a legnagyobbat bukni is. Különben is, fabatkát sem ér a legnagyszerűbbnek tűnő elmélet sem, ha a gyakorlatban nem válik be. A szakmával kapcsolatos véleményemet is saját gyakorlati tapasztalataimból alakítottam ki. Viszont hangsúlyozom, hogy jó gyakorlatot csak megalapozott elméleti ismerettel lehet végezni.

– Ilyen szempontból a Primo pályafutása hozzájárult ennek a véleményének alakulásához? Annak idején nagy feltűnést keltett a Primo bemutatása, csupán az a tény is, hogy az akkor kapható hasonló gépekhez képest feleannyiba került.

– Tönkre is ment néhány, ezzel a gépkategóriával foglalkozó kisvállalkozás.

– Hogyan sikerült nulláról indulva, alig másfél év alatt a Primót piacra dobni, ráadásul ilyen olcsón?

A mikro-számítástechnika nem mikroszintet jelent!

BESZÉLGETÉS MÓRICZ SÁNDORRAL

– A Primo történetéhez tartozik, hogy előbb-utóbb venni akartam gyerekeimnek egy számítógépet, és körülnéztem a piacon; láttam, hogy mi van, illetve mi nincs. Engem mindig érdekelt az innováció egész folyamata, ideértve azt is, hogyan lehetne a gyártó és a felhasználó közötti űrt betölteni. Ehhez korábban készítettem egy elméleti modellt, ezt most elővettem az íróasztalból, és részletes vizsgálat alapján kidolgoztam a konkrét paramétereket. Ezután már „csak" össze kellett szedni egy fanatikus csapatot, amelylyel sikerre lehetett vinni az ügyet.

– De hogyan tudták az egészet ilyen rövid idő alatt dűlőre vinni? Arra gondolok, hogy egyesek szerint nálunk baj van a szerződéses fegyelemmel.

– Ez tény. Ezzel munkatársaimmal együtt számolnom kellett. Olyan menedzselési stílust kellett kialakítanunk, amelynél az alvállalkozókat az anyagbeszerzőtől a vezérigazgatóig állandó folyamatos stresszállapotban tartva, kiverekedtük a határidők betartását. Ez persze nekünk is rettenetesen fárasztó volt, különösen ha meggondoljuk, hogy normális „üzemmenetnél" ez teljesen felesleges lett volna.

– A Primónál maradva: milyen általánosítható tapasztalatokat szerzett a gép pályafutása során?

– Előbb inkább a konkrétat említem, mert ez biztató a jövőre: sikerült egy olyan szemléletet kialakítani, hogy például, amikor egy bizonyos műszaki megoldás került szóba, az egyik munkatársam, aki egész életében hardverrel foglalkozott, közbeszólt, hogy ezt másképp kellene megoldani, mert az alkatrész beszerzése a végső árban ezreket jelenthet! Visszatérve a kérdésre, vannak általánosítható tapasztalatok. Például itt van mindjárt a gazdasági környezet, a számítógépeket érintő irányítás. Minden korban, minden országban a húzó iparágakat preferálják, az egyes vállalatokat különböző kedvezményekkel terelik a helyes irányba. A Primót és más mikrogépeket viszont indokolatlanul magas adó sújtja. Egy 10 ezer forintos gépnél 26 százalékos adó 2600 forintot jelent. Tessék a maradék 7400 forintból kigazdálkodni a drága alkatrészeket, munkabéreket és minden más egyebet az eredetileg kitűzött áron, ilyen „piti" szériák mellett! Munkatársaimmal együtt 20 fillérekért veszekedtem a gyártóval, pedig régi, bevált partner.

– Hallottam olyan nézetet, hogy a jelenlegi szabályozók nem kis szerepet játszanak abban, hogy rengetegféle gép van forgalomban, ami alapjaiban gátolja az egységesítést, megakadályozza egy igazi szoftverpiac létrejöttét.

– Ezt a nézetet magam is osztom, és a felhasználók oldaláról nézve olyannak látom a helyzetet, amit azzal a bizonyos piroslámpás házzal szoktak jellemezni. És ha meggondoljuk, hogy a fejlesztések heterogén gépparkra történnek, még nagyobb a kuszaság. Egy vállalat vesz például egy gépet, és ebbe a sajtó is belehajszolja, hogy ettől várhatja minden problémája megoldását. Megbíz egy kisvállalkozót, hogy dolgozza ki rá az anyaggazdálkodását. Elkészül a program, mondjuk, megy is, de azon az egy gépen futtatható csak, az idők végezetéig. Ez képtelen pazarlás.

– Igen, de mit lehetne tenni, hogy ne így legyen? Gondolom, mindenki tudja, hogy nem jó ez így. Miért nem ülnek le és állapodnak meg, hogy kompatíbilis gépeket választanak ki, és a jövőben ezekre történik a fejlesztés?

– Ezerszer összeültek már, de a szép szavak papíron maradtak. Konkrét tettek kellenek!

– És az irányítás? Nem lehetne gazdasági szabályozókkal megoldásra juttni?

– Csakis azokkal lehet. Mikor a jelenlegi ötéves tervet készítették, még nem tudhatták, hogy ilyen intenzív lesz a szakma felfutása. Bízom abban, hogy az új ötéves tervbe már ezt is beszámítják. Különben marad a mostani helyzet, hogy mindenki a saját dombján kukorékol, azaz a beszerezhető alkatrészekből összeállít egy gépet, hozzátesz még valamit, hogy egyrészt magyar, másrészt saját legyen, aztán futtassa, aki tudja. Hisz abban, hogy ő és csakis az ő megoldása zseniális. Ha a felhasználónál nem válik be, a magyarázat általában az, hogy nem képes felfogni az ő zsenialitását. Pedig a szellemi nagyságnak olyannak kell lennie, hogy a megoldások mindenki számára érthetők legyenek! A számítástechnikusoknak le kell tudni ereszkedni a mindennapok problematikájához, tudniuk kell kommunikálni a mindennapok gyakorlatában szenvedő emberekkel, különben soha nem fognak sikeres alkalmazásokat létrehozni. Sokkal nyitottabbnak kell lenni!

– Hogyan látja a hazai mikro-számítástechnika jövőjét?

– Talán eretnek nézetnek hangzik, de szerintem a jelenleg folyó hardverfejlesztések távlatilag és gazdaságilag nem kecsegtetnek eredménnyel. Talán egy részükkel fel kellene hagyni. Az erőt, energiát és pénzt egy egységesített gépparkra alapított továbblépésre kellene fordítani. Mondjuk úgy, hogy a kommersz gépeket, amelyeket itthon másfél millió forintért tudunk piacra hozni, egyszerűen Hongkongban vagy másutt 2–300 ezer forintnak megfelelő valutáért szereznénk be, és amiket csak össze kell szerelni. Foglalkozni kell a mikrogépekkel, de nem Commodore 64-re alapuló kisvállalkozói mikroszinten. A kisvállalkozások sem eszközeikkel, sem tudásuknál fogva nem képesek egy nagyobb vállalati rendszert átfogni. Egy csúcsmenedzsment irányításával az egységes gépekre olyan hálózatot lehetne kialakítani, amelyben az azonos feladatokra – például a raktározás lényege mindenhol ugyanaz – azonos alkalmazásokat lehet kifejleszteni, és a gépeket az adott vállalat nagyságrendjének megfelelően lehetne a hálózatra kapcsolni.

LACZKA MIKLÓS

Képletkezelő programok

Mit tud egy képletkezelő program? Egyszerűsíteni, összevonni, behelyettesíteni, több tagot szorozni több taggal, alkalmazni a nevezetes és a trigonometrikus azonosságokat, differenciálni és esetleg integrálni is, tehát azokat a műveleteket képes elvégezni, amelyeket régebben a középiskolában algebrának neveztek.

Lássunk egy alkalmazási példát. Keressük egy ellenálláshálózat *(1. ábra)* eredő ellenállását.

$$R = \cfrac{1}{\cfrac{1}{R_1} + \cfrac{1}{R_1 + \cfrac{1}{\cfrac{1}{R_2} + \cfrac{1}{R_3}}}}$$

Az R-re kapott képletet begépeljük programunknak:

```
R := 1/(1/R1 + 1/(R1 + 1/(1/R2 + 1/R3)))
```

A program válasza:

```
( R1**2*R2 + R1**2*R3 + R1*R2*R3 ) /
( 2*R1*R2 + 2*R1*R3 + R2*R3 )
```

(Itt a hatványozást – a FORTRAN nyelvhez hasonlóan – két csillag jelöli.)

Ha most tudjuk, hogy például R1 = 0.5 ohm és R3 = 2*R2, ezt beadhatjuk a programnak, majd megkérdezzük R új értékét, tehát begépeljük:

```
R1:=1/2; R+:=2*R2; R
```

A válasz: (3 + 4*R2) / (12 + 8*R2)

Egy program megírásakor két dolgot döntünk el: az adatok ábrázolási (tárolási) és feldolgozási módját. Azaz meghatározzuk a megfelelő adatszerkezetet és a hozzá illő algoritmust.

Adatszerkezet: a kifejezések ábrázolása fával

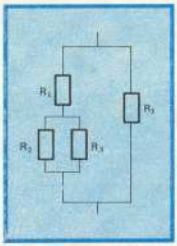
A BASIC nyelvben a kifejezéseket egy karaktersorozattal adjuk meg. Ez hasonlít a képletek szokásos matematikai írásmódjához. Az ilyen felírást infix formának nevezzük, mert a műveleti jelek mindig a számok vagy változók között állnak.

A FORTH nyelv esetén is karaktersorozattal adjuk meg a kifejezéseket (ezeket a FORTH kétbájtos címek sorozataként tárolja), de itt például a 2+3+4-et így kell írni: 2 3 + 4 +, vagy 2 3 4 + +. Ez a postfix forma, a műveleti jel (operátor) mindig a műveletben részt vevő értékek (operandusok) után áll.

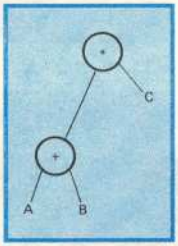
Olyan nyelv is létezik – például a LISP –, ahol az operátort mindig az operandusok elé kell írni, például + 2 3. Ez a prefix forma.

Kifejezéskezelésre ezek egyike sem ideális, mert nem tükrözik a képletek szerkezetét, ezért célszerűbb a fa szerkezetű ábrázolást választani.

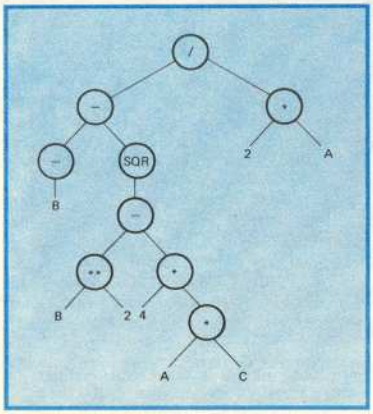
A fának ágai vannak, az ágakból újabb ágak eredhetnek, azokból még további ágak, az ágak végén pedig levelek vagy gyümölcsök ülnek. Egy matematikai képlet abban hasonlít egy ilyen fára, hogy mindkettő „önmagába skatulyázott szerkezetű". A leveleknek a képletben szereplő számok és betűk felelnek meg.



1. ábra



2. ábra



3. ábra

Az (A+B)*C képletnek megfelelő fát mutatja a *2. ábra*. Ez egy furcsa fa, mert lefelé nő, de így szokták rajzolni. Bonyolultabb példa a *3. ábra*.

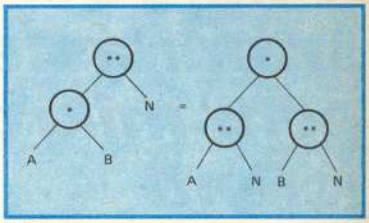
$$\frac{-B-\sqrt{B^2-4\cdot A\cdot C}}{2\cdot A}$$

A fa ábrázolásból egyszerű algoritmussal visszakapható a képlet BASIC vagy FORTH alakja.

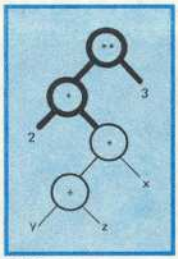
Algoritmus: a helyettesítés

A helyettesítés a kifejezés-fák kezelésének alapművelete. Például adott a (2*(Y+Z)*X)**3 kifejezés, ezt szeretnénk átalakítani az (A*B)**N = A**N*B**N nevezetes azonosság segítségével. Ezt megtehetjük, mert olyan a képlet szerkezete, mint az azonosság bal oldaláé, csak A helyett 2, B helyett ((Y+Z)*X), N helyett 3 áll.

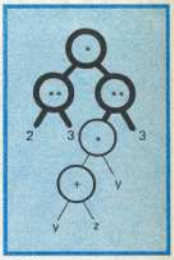
A *4. ábra* az azonosságot, az *5. ábra* pedig a próbakifejezést szemlélteti. Látjuk, hogy az azonosság bal oldalát ábrázoló fa ráilleszthető a próbakifejezést ábrázoló fára (vastag vonallal kihúzott rész), A-nak megfelel 2, N-nek 3, B-nek pedig az ábrán nem vastagított rész.



4. ábra



5. ábra



6. ábra

Ha ezeket beírjuk az azonosság jobb oldalába A, B, N helyére, akkor már meg is kaptuk a próbakifejezés átalakított formáját *(6. ábra)*, azaz 2 **3 * ((Y+Z) *X) **3-at.

Erre az eredményre még egyszer alkalmazhatjuk az azonosságot, az azonosság bal oldala a kifejezés egy részfájára illeszkedik *(7. ábra)*. A-nak az ábrán külön kiemelt rész, B-nek X, N-nek 3 felel meg. A végeredményt a *8. ábra* mutatja.

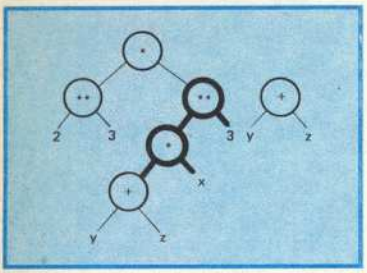
Néhány elnevezés: a helyettesítéshez használt azonosságot itt helyettesítési szabálynak hívjuk, utalva arra, hogy csak az egyik irányban kívánjuk használni (el kell dönteni, hogy éppen beszorozni vagy kiemelni akarunk, különben a program oda-vissza ismételné a két műveletet a végtelenségig). A helyettesítési szabályban szereplő betűket (a szabályt ábrázoló fa leveleit) formális paraméternek nevezzük, az ezeknek megfelelő részfák a próbakifejezésben az aktuális paraméterek.

Megjegyezzük, hogy egy összeget többféleképpen rajzolhatunk le fával, mivel az összeadás műveletét hallgatólagosan kétargumentumúnak ábrázoltuk (a BASIC és FORTH nyelveknél is mindig két operandusra érvényes a „+" jel). Például az A+B+C+D kifejezést a *9. ábrán* látható változatok bármelyikével ábrázolhatjuk. Az egy-egyértelmű ábrázolás az lenne, amit a *10. ábra* mutat.

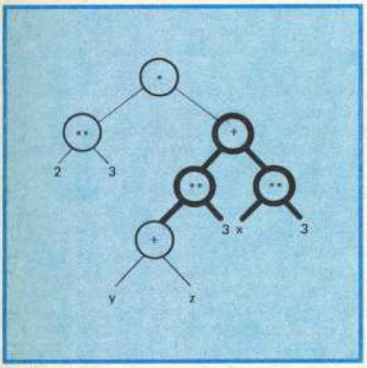
Az összeadásnak azt a tulajdonságát, hogy (A+B)+C = A+(B+C), asszociatív törvénynek nevezzük. Csakhogy ha így ábrázonánk az összegeket és szorzatokat, elbonyolódna a helyettesítés definíciója, ezért most maradunk az eredeti, kétargumentumos megoldásnál. A fának ezt az önkényes (csak az ábrázolásból adódó) szerkezetét a FORTH (postix) forma egyértelműen megadja, a BASIC (infix) formában pedig zárójelezéssel írhatjuk le.

Például az előbbi fák esetén így:

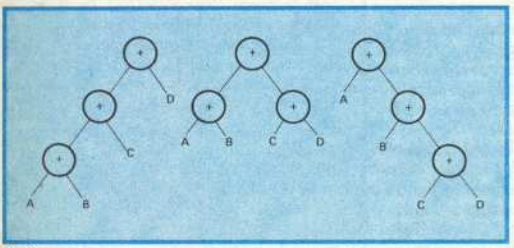
((A+B)+C)+D	A B + C + D +
(A+B)+(C+D)	A B + C D + +
A+(B+(C+D))	A B C D + + +



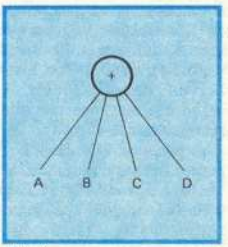
7. ábra



8. ábra



9. ábra



10. ábra

A BASIC nyelv – ha nem használunk zárójeleket – az első formának megfelelő sorrendben végzi a műveleteket, a balról jobbra szabály szerint. Mi is ezt a feliratot használjuk a következő példában.

Tekintsünk egy nagyon egyszerű „algebrát", amelyben csak összeadás van, azaz a kifejezések csak számokból (pozitív vagy negatív számokból), betűkből és „+" jelből épülhetnek fel. Ilyen kifejezés például:

$-1 + B + A + 2$ azaz: $((-1+B)+A)+2$

Ennél az „alig-algebránál" a kifejezések feldolgozása két dolgot jelent: a változók ABC sorrendbe rendezését és a számok összevonását. Megadjuk azokat a helyettesítési szabályokat, amelyek elvégzik ezeket:

1. $X+Y \rightarrow Y+X$ ha X szám vagy betű Y is szám vagy betű és $X>Y$
2. $(X+Y)+Z \rightarrow (X+Z)+Y$ ha Y szám vagy betű Z is szám vagy betű és $Y>Z$
3. $N+M \rightarrow val(N+M)$
4. $(X+N)+M \rightarrow X + val(N+M)$
5. $X+\emptyset \rightarrow X$

Az 1. és 2. szabály biztosítja a rendezést. A betűk legyenek „kisebbek", mint a számok (az ASCII kódnál például fordítva van, a számok kódja a kisebb), így a helyettesítés a kifejezés végére rendezi a számokat. A 3., 4., 5. szabályok végzik a számok összevonását, N és M (pozitív vagy negatív) szám lehet. A „val" a kifejezés számértékét jelenti.

Próbakifejezésünk átalakításának lépései a következők:

$((-1+B)+A)+2 \rightarrow ((B+-1)+A)+2$
1. szerint
$((B+-1)+A)+2 \rightarrow ((B+A)+-1)+2 \rightarrow$
$((B+A)+2)+-1$
2. szerint
$((B+A)+2)+-1 \rightarrow ((A+B)+2)+-1$
1. szerint
$((A+B)+2)+-1 \rightarrow (A+B)+1$
4. szerint

Többféle művelet, bonyolultabb szabályrendszer esetén is meg lehet próbálni a kifejezésfeldolgozó algoritmust helyettesítési szabályok segítségével megadni. De ez azzal jár, hogy bonyolultabbá válnak a szabályok alkalmazásának feltételei, és az algoritmus nagyon lassú lesz, mert minden kifejezésnél és annak minden részkifejezésénél végig kell vizsgálni az összes szabály alkalmazhatóságát.

Ezért a gyakorlatban közvetlen algoritmusokat használnak a sűrűn előforduló műveletekre: a rendezésre, összevonásra, polinomok osztására, differenciálásra stb. Ezek az algoritmusok „behuzalozva" tartalmazzák az alapműveletek tulajdonságait (a nevezetes azonosságokat), így hátrányuk a helyettesítéses módszerhez képest az, hogy merevebbek annál.

A rendszerek egy része úgy dolgozik, hogy minden kifejezést egy egységes „kanonikus" formára alakít (például a betűket mindig ABC-be rendezi, a hatványokat nagyság szerint, a zárójeleket felbontja, a törteket közös nevezőre hozza stb.). Két kifejezést akkor vesz egyenlőnek, ha az egységesítés után a kifejezéseket egyforma fa ábrázolja: összevonásnál és egyszerűsítésnél az egyenlő kifejezések különbsége nulla, hányadosa egy.

A működő formulamanipulációs rendszerek

Egy egyszerűbb képletkezelő program az alábbi részekből áll:

– Kifejezések beolvasása és kiírása, a belső ábrázolás (fa) és külső ábrázolás (karaktersorozat) közötti átalakítás. Kényelmes, ha a külső ábrázolás közel áll a szokásos matematikai írásmódhoz (törtvonal, alsó-felső indexek).

– Helyettesítő algoritmus, változtatható helyettesítési szabályokkal.

– Képletegységesítő algoritmus. Ennek is többé-kevésbé befolyásolható a működése: „kapcsolók" állításával meg lehet adni, hogy fel akarjuk-e bontani a zárójeleket, ki akarjuk-e fejteni az összegek hatványait stb.

Néhányat felsorolunk a bonyolultabb rendszerek lehetőségei közül:

– Számítási program tárolása. Egy BASIC-hez vagy Pascalhoz hasonló nyelven le lehet írni a feladat programját; a programban szereplő változók értéke vagy egy kifejezés (fa), vagy egy szám. A számok lehetnek egészek, törtek és lebegőpontos számok. Általában több száz jegyű egész számok is alkalmazhatók.

– Vektorok, mátrixok.

– Egyenletek, egyenletrendszerek megoldása.

– Integrálás.

Milyen nyelven célszerű képletkezelő programot írni? Jó, ha a nyelv tud fa szerkezetekkel és rekurzív szubrutinokkal dolgozni. A legelterjedtebb ilyen nyelv az 1960-ban kidolgozott LISP (LISt Processor). A futási idő és a szükséges memória nagymértékben függ az alkalmazott nyelvtől és algoritmusoktól.

Az első formulamanipulációs programokat már az ötvenes évek végén megírták. A fejlődés fő ösztönzője a fizikai kutatás volt, többek között az égi mechanika és a relativitáselmélet. Sokszor előnyösebb egy eredményt képlet formájában, mint numerikusan (számok, ábrák alakjában) megkapni, főleg, ha az eredmény több változótól függ. Ezenkívül a numerikus számítást mindig terhelik a kerekítésből adódó hibák, a hibák hatása pedig néha drasztikus, és nem is mindig becsülhető meg pontosan.

A képletkezelés felhasználja a formális logika eredményeit, szegről-végről rokonságban van a tételbizonyító és programhelyesség-bizonyító rendszerekkel is.

A nagyszámítógépekre több formulamanipulációs program is létezik: REDUCE, FORMAC, MACSYMA stb. Ezek egy része hazai számítóközpontokban is hozzáférhető. A gyakorlatban is érdekes feladatok esetén a tárigény több száz kbájt, a futási idő több perc vagy több óra. Kézzel, papíron a megoldás általában évekig tartana.

A nagygépes rendszerek legtöbbször nem interaktívak. A „nagygép" kifejezés persze relatív. A személyi számítógépek napjainkban elérik azt a teljesítményt, ami a komolyabb képletkezelő rendszerek futtatásához kell (16 bites mikroprocesszor, 256 k memória, 10 MHz órajel stb.). De egy egyszerűbb rendszer akár egy C64 gépen is futhat, CP/M-re is létezik ilyen program. A formulamanipulációnál nagy előny az interaktivitás, mert az ember sokszor csak a képlet alakítása közben jön rá, hogy milyen műveleteket lenne érdemes kipróbálni.

Feladatok

1. Hogyan célszerű ábrázolni a fa szerkezetet a számítógép memóriájában?
2. Írjuk meg a kifejezéseket fává (és vissza) alakító programot!
3. Írjuk meg a helyettesítő algoritmust!

MÁRK GÉZA

Görbére illeszkedő polinom kiszámítása

Tudományos számításoknál olykor szükség van olyan polinomra, amely illeszkedik arra a függvényre, amelyből csak néhány pontot ismerünk. Az alábbi program egy olyan polinom együtthatóit számítja ki, amely az általunk megadott (x_i; y_i) pontokra pontosan illeszkedik.

Ha p+1 adatunk van, akkor az a görbe, amely az összes (p+1) pontot tartalmazza, legalább p-ed fokú polinomot eredményez. A program ennek együtthatóit számítja ki, és néhány speciális esetben ezek megegyeznek az összes pontot tartalmazó egyetlen – alacsonyabb fokszámú – polinommal is.

A program a Lagrange-féle interpolációs polinomon alapul. Eszerint minden x_i (ahol $0 < i < p$; egész szám) abszcisszaértékhez előállítunk egy olyan p-ed fokú $A_i(x)$ polinomot, amelynek két jellemző tulajdonsága:

x_i helyen értéke 1,

minden egyéb x-re a $0 \le i \le p$ tartományban 0 értékű.

Az $A_i(x)$ polinomot két lépésben állítjuk elő. Először a (130–190 sorok között) $F_i(x)$ polinomot számítjuk ki, ahol

$$F_i(x) = (x - x_0)(x - x_1)\ldots$$
$$\ldots(x - x_{i-1})(x - x_{i+1}) \ldots (x - x_p)$$

Ez csupán a második tulajdonságot teljesíti, azaz 0 értéket ad $F_i(x)$-nek (a 220–230-as sorban) – az x_i-től különböző helyeken.

Ezután előállítható az $A_i(x)$ függvény (a 200-as sorban), amely már teljesíti az említett két tulajdonságot:

$$A_i(x) = F_i(x)/F_i\ (x_i \text{ értéknél})$$

A kívánt függvényhez kiszámítunk egy-egy $A_i(x)$ értéket minden x_i-re, megszorozzuk az Y_i értékkel az x_i-hez tartozó $A_i(x)$ együtthatóit, majd a polinomokat összegezzük. A program ezt a 275–315 sorban teljesíti, majd az együtthatókat a csökkenő hatványok sorrendjében kiírja.

A program az x = 0, 1, 2, 3, ... értékhez tartozó adatokat összegezi, de megenged más egyforma egységekre osztott skálázást is. A 160-as sor átírásával a program egyenlőtlen közökkel felvett adatokra is alkalmazható.

Az $x_0 - x_p$ pontokon kívüli tartományban a közelítő polinom hibája – a pontok közötti ingadozáshoz képest – általában rohamosan növekszik.

A program Homelab–2 (Aircomp) számítógépen futott, de minimális átalakítással más készülékekre is alkalmazható. Minthogy a Homelab–2 aritmetikájának pontossága a 6 kijelzett számértéknek felel meg, és a polinom-előállítás során a legvalószínűbb kerekítési-számítási hibát a szorzás művelete okozza, ez megszabja a beadható pontok számának célszerű határát.

A program maximum 15 pont adatait fogadja, viszont minden pont a szorzás miatt legalább egy nagyságrend növekedést okoz a számítások során, így 6 pont fölött a magasabb kitevőjű együtthatók értékét bizonyos kritikával kell fogadni.

Az eredmények gyorsabb elérése érdekében a számítások idejére a program – a DL értékének megfelelően – lekapcsolja a képmegjelenítést.

Példaként az $Y = x^2 - 6x + 10$ függvény értékeit beadva, a program a következő együtthatókat számította ki:

J(5) = −1.86265E−8, J(4) = 2.68221E−7,
J(3) = 0, J(2) = 1, J(1) = −6, J(0) = 10

(A cikk az Electronic Engineering 1985. februári cikke nyomán készült.)

TOLNAI GÁBOR ISTVÁN

```
 50 A=A' "ILLESZKEDO POLINOM
    KISZAMITASA"
100 DIM A(16,15), C(16), H(15), Z(15)
115 INPUT "HANY PONTRA ILLESZTUNK?
    (0<P<15) ";P:P=INT(P)
120 IF P>15 OR P<1 THEN GOTO 115'
    GOSUB 245
125 FOR J=0 TO P
130 GOSUB 235 ' "C(I) NULAZASA"
135 C(0)=1:DL=0
140 FOR N=0 TO P
145 R=N+1
150 IF N=J THEN N=N+1
155 IF N>P THEN GOTO 190
160 S=-N
165 FOR I=1 TO R
170 A(I,J)=S*C(I)+C(I-1)
175 NEXT I
180 A(0,J)=S*C(0)
185 FOR I=0 TO R: C(I)=A(I,J): NEXT I
190 NEXT N
195 GOSUB 220
200 FOR I=0 TO P: A(I,J)=A(I,J)/C(0)
    : NEXT I
205 NEXT J: DL=200
210 GOSUB 275' "EREDMENY KIIRATAS"
215 PRINT "KESZ!": END
220 FOR I=P TO 1 STEP -1
225 C(I-1)=C(I)*J+C(I-1): NEXT I
230 RETURN
235 FOR I=0 TO P+1: C(I)=0: NEXT I
240 RETURN
245 PRINT: PRINT "BEMENO ADATOK"
    : PRINT"=============="
    :PRINT
250 FOR I=0 TO P: PRINT I;
255 INPUT "ABSZCISSZAHOZ TARTOZO
    'Y'-ERTEK:"; Z(I)
265 NEXT I
270 RETURN
275 PRINT: PRINT
    " AZ ILLESZKEDO POLINOM:":PRINT
280 FOR I=0 TO P: H(I)=0: NEXT I
285 FOR J=0 TO P
290 FOR I=0 TO P
295 H(I)=Z(J)*A(I,J)+H(I)
300 NEXT I
305 NEXT J
310 FOR I=P TO 0 STEP-1: PRINT
    "J(";I;")=";H(I),: NEXT I
315 RETURN
```

Egy számítógépes fertőzés kórtörténete

1. A számítógépé a jövő. Erről a házastárs sikeres meggyőzése. Gyermekek jövőjének említése. Külföldi út. Megtakarított valuta. Olcsó személyi számítógép vásárlása. (Programkazettákra, tartozékokra már nem futja.)

2. A laikusok csodálkozása: „Ez egy számítógép?" „A program a tévében van?" „Közönséges magnó?" stb. Rövid magyarázat. Figyelmük felhívása a TV-BASIC-re.

3. Játékprogramok felhajtása. Másolóprogramok beszerzése. A család a képernyő előtt. Találat esetén indiánüvöltés, visítás. Hatás: vacsora sebtében, rendrakás szintén. Gyereklefektetés drasztikus módszerekkel. A felnőttek tovább játszhatnak. Másnap karikás szemek, ásítás.

4. A hétvége a számítógépé. Takarítás elmarad. Mosás szintén. Evés, leckeírás sebtében. Görnyedt hátak, égő szemek. Elég volt! Megálljatok! A gép nemcsak játékszer! Dolgoztatni is tud!

5. Géptől távol tartó programok írása: szorzótábla-kikérdező a hétévesnek, kis összeadások az ötévesnek. Találat esetén jó pont, lágy zene. Hiba esetén fekete pont, fals zene. Végén értékelés. Bonyolult matematikai játék a házastársnak.

6. Lehetetlen! Csoda! A gép még jobban vonz! Szorzótábla hamar megtanulva. A középsős óvodás már összead. Borzasztó! Újabb oktatóprogramokat kell írni!

7. A gépnek már híre van. Baráti, gépnéző látogatások. A gépmánia terjed. Iskolatársak lelkesedése, a billentyűzet maszatolódása. Új programokat követelnek! Tévémaci helyett TV-BASIC. Mindenki BASIC parancsokkal álmodik. Meddig lehet még ezt bírni? Krízis. Segítség! Terápiát kérünk!

KÖRÖSNÉ MIKIS MÁRTA

Akik csatát veszíthetnek, de háborút nem

A személyi számítógépes technika legújabb vívmányainak tömeges elterjesztésében kiemelkedő szerepet játszanak az angolul *maverick*-nek nevezett, nagy vállalkozói egyéniségek. A szó tulajdonképpen lefordíthatatlan. Azt jelenti, hogy az ipar belső berkeiben kialakult aktuális közgondolkodástól néha homlokegyenest eltérő, netán szélsőségesen különcnek tűnő megközelítéssel lépnek fel ezek a mindentől és mindenkitől független, nagy vállalkozók.

Nem kalandorokról van szó, bár néha kalandozásnak tűnhet mindaz, amit véghezvisznek. A lehető legjobb ismerői ugyanis a technológiai tendenciáknak és a lappangó, még fel nem tárt piaci lehetőségeknek. Látnoki erővel tudnak elvonatkoztatni a pillanatnyi technikai és piaci helyzettől. Magas intelligenciájukkal valamennyi fontos tényezőt egy személyben át tudnak tekinteni és a következő, a korábbitól esetleg merőben eltérő vállalkozásuk középpontjába állítani. Ebben közvetlen környezetük, de még egymás véleménye sem befolyásolhatja őket.

Kik is ezek a jelenkori szellemi munka intenzív korszakának nagy lovagjai?

Clive Sinclair

Egyikük valóságos lovag. Az angol királynő ütötte azzá, sikerei elismeréseképpen. Sir Clive Sinclairről van szó. Karrierje pedig egyáltalán nem indult valami fényesen. Iskoláit 17 éves korában befejezve nem tanult tovább, hanem belevetette magát az élet sűrűjébe. Először a Practical Wireless újságírójaként dolgozott, majd 22 éves korában megalapította első cégét, a Sinclair Radionicsot. A cég rádió és hifikészleteket (kiteket) gyártott, amelyeket postai csomagküldéssel szállított a vevőknek. Később kalkulátorok és digitális órák gyártásába is belekezdett. 1976-ban azonban a csőd szélére került, mivel nem bírta a versenyt a jó minőségű tömegáruval megjelenő japán konkurrenciával.

Sinclairt a brit kormány húzta ki a csávából. A pénzügyi „injekció" lehetővé tette, hogy folytassa úttörő elektronikai munkásságát. 1979-ben vitája támadt a kormányérdekeltséget ellenőrző National Enterprise Boarddal, és kilépett a cégtől. Két igen jelentős mikroszámítógép-konstrukciót kellett otthagynia, csak a mindenki által játékgépnek tekintett ZX80-tervet vihette magával. Júliusban megalapított új vállalata, a Sinclair Research hamarosan piacra hozta a korábbi nevetség tárgyát, a ZX80-at, mégpedig mindössze 100 fontért. Furcsa módon 100 ezer darabot sikerült eladnia valamivel több, mint egy év alatt ebből az 1 kbájtos masinából. Azt mondanunk sem kell, hogy az otthagyott mikrogépekből gyakorlatilag semmi sem lett.

1981-ben kihozta a javított, ZX81 konstrukciót, majd 1982 nyarán a már igazán nem játékgép szintű ZX–Spectrumot. És micsoda árakon! 1983 szeptemberében 40 font körüli összegért árulták a ZX81-et, a 48 k-s Spectrumot pedig 120 fontért. Titka nagyon egyszerű: a ZX81 előállítási ára kb. 5–10 font volt, a Spectrumé pedig 16–20 font. „Csúcsteljesítményem mindössze abban áll, hogy 10 pennyért tudom megcsinálni azt, amit egy fontért minden bolond képes előállítani" – magyarázta ebben az időben amerikai stílusú üzleti filozófiáját Sinclair.

A Spectrum piacra hozatalakor „még csak" 500 ezres gépparkot maga mögött tudó Clive bácsi (jelenleg 44 éves) az új géppel több milliósra növelte eladott gépeinek állományát. Ennek nagy részét, kb. 70 százalékát exportálta. Sok országban neki köszönhető, hogy nagy tömegek igen rövid idő alatt megismerkedhettek a számítógépes technikával. Hazájában pedig neki tulajdonítják, hogy a világ élenjáró számítógépes kultúrájú nemzetévé válhattak.

Jack Tramiel

Sinclair fő erényének tartják, hogy mindenkinél gyorsabban képes eligazodni az új alkatrész- és szerelési anyag választékban, továbbá, hogy mindenkinél bátrabban alkalmazza az innovatív, berendezésorientált integrációs technikákat. Egy ponton azonban sok benne a közös Jack Tramiellel, a Commodore volt elnökével: mindketten az „add olcsón és halmozd fel magasra" (sell it cheap and pile them high) mottójú, jól bevált értékesítési koncepciót követik. A tömeges fogyasztásra szánt elektronika tipikus vásárlója ugyanis impulzív természetű vevő, akit semmi sem befolyásol jobban, mint az „ennyire olcsó és ily sokan vásárolják" érzése.

Tramiel azonban a „big business"-szel való szembeszállás nagy bajnoka. Nem a korszerű elektronika nagy tömegek számára elérhető, innovatív alkalmazása a legfőbb motivációs tényező számára, hanem főként az, hogy megmutassa: ő, Jack Tramiel többre képes, mint a gondos vezetői munkával olajozott, hatékony működésűvé tett, nem kis tőkeerőt képviselő, nagy szervezetek. A Texas Instrumentset, amely korábban sok borsot tört az orra alá, 1983-ban „kapta el".

A Texas a korábbi kalkulátorpiaci akció után az éppen intenzív felfutásban levő háziszámítógép-piacot akarta megkaparintani 1981 közepén piacra hozott 99/4A gépével. A gépbe korábbi egyszerű miniszámítógépének 16 bites és 64 kbájtos architektúráját alkalmazó mikroprocesszorát építette be. Stratégiájában a szolid, megbízható szoftver- és hardverkiegészítők biztosítására is nagy súlyt helyezett. Egyszóval a piacot minőségben és árban meghatározó tényezővé próbálta magát tenni, hogy hosszú távú vezető szerepe is érvényesüljön.

Tramiel szintén 1981 nyarán jelent meg konkurrens gyártmányával, a VIC–20-szal. Ez tulajdonképpen egy sokkal alacsonyabb áron „újracsomagolt" PET volt. Fő előnyének bizonyult a későbbiek során, hogy nemcsak elődjének sikerére támaszkodhatott, hanem harmadannyi integrált áramkört tartalmazott, mint a Texas gépe. 1982 őszén, amikor mindkét gép listaára 299 dollár volt, a VIC–20-at 230–250 dolláros tényleges piaci áron vették, mint a cukrot, a 99/4A-ra pedig 100 dolláros árengedményt kellett bejelentenie a Texasnak, hogy 40–50 ezer darabra felhalmozódott készleteit apassza. Ezzel megnyitotta a számára hamarosan végzetessé váló árháborút.

Tramiel gondolkodás nélkül vállalta az árharcot. A Texasnak végül már 99 dollárra kellett csökkentenie gépe árát, miközben csak maga az előállítási ár 15–20 dollárral meghaladta ezt az összeget. A VIC–20-at ekkor 89 dollárért árulták, ami még mindig több volt, mint a becslések szerint 60 dolláros gyártási önköltség.

Az első árengedménynél a Texas még arra gondolt, hogy a szoftver- és a periféria-eladásoknál pótolja majd a központi egységnél elszenvedett veszteséget. De egyik elképzelése sem jött be. Mert ugyan ki vesz egy 99 dolláros géphez 400 dolláros floppymeghajtót? Szoftverei árát a Commodore 20 dollár alá tudta csökkenteni, míg a Texas 30 és 40 dollár közötti programjai már azért sem voltak kelendőek, mert elsősorban oktatási célúak voltak, márpedig egy 99 dolláros gépet elsősorban játékprogramok futtatására vásárolnak a vevők.

A több milliárdos éves forgalmat bonyolító elektronikai óriás 100 millió dollár negyedéves veszteséget volt kénytelen elkönyvelni 1983 júniusában. Hamarosan ki is szállt mindörökre a háziszámítógép-üzletből. A finálét a kiárusított készletek szuperalacsony eladási ára kísérte, amikor a VIC–20-akat is már gyártási és beszerzési ár alatt árulták a kiskereskedelemben és az áruházakban, csak hogy szabaduljanak tőlük. Tramielnek még ez sem okozott gondot, mivel időközben felfuttatta az 1982 nyarán 595 dolláros listaáron bejelentett Commodore 64-et, és az igényesebb vevők részére egy számára is kedvező vásárlási alternatívát tudott ajánlani a kritikus időszakban. A karácsony előtti vásáron már 200 dollárért is hozzá lehetett jutni a C64-eshez, ami még fokozta ez iránt a jól sikerült házigép iránt a keresletet. A 90 dolláros előállítási ár mellett még így is óriási nyereség volt a gépen.

A piaci pozíció szempontjából döntő USA mellett Tramiel igen nyereséges világpiacot is kialakított magának. Termelésének közel a felét értékesítette az USA-n kívül. 1980-ig még Japánban is vezetett PET gépével az eladások tekintetében. Az 1983-as évet 1 milliárd dolláros forgalommal záró Commodore már majd négymillió számítógép eladását tudhatta maga mögött, és saját piacán a vezető világcég lett.

NINO

(Folytatjuk)

A technológiák szerepe

Az Apple cég az eredeti személyiszámítógép-gyártók között vezető helyet szerzett magának, hála a nyílt és bővíthető konstrukciónak, a megjelent és innovatív alkalmazási termékeknek és az új cég sikeres vállalatvezetési gyakorlatának. „Mindig olyan embereket igyekeztünk felvenni, akik nálunk jobbak voltak az adott területen" – nyilatkozta 1982 végén Steve Jobs. „A kulcs a vegyületek megfelelő egyensúlya."

Az első 1000 kártya legyártásához 250 ezer dollárra volt szükség. Ismerősük, Mike Markkula fantáziát látott a dologban, megvolt a megfelelő üzleti felkészültsége, így betársult a két fiúhoz. Elkészítette a vállalkozás átfogó üzleti tervét. Irányításával az Apple mint rendezett jogi és pénzügyi alapokon működő vállalat kezdte meg 1977 szeptemberében első teljes évét. Egy év múlva már nyereséget fizetett részvényeseinek, igaz, hogy mindössze 3 dollárcentet részvényenként. Ezt a hozadékot azonban minden évben jelentősen növelte.

A fokozódó siker mindig lehetővé tette, hogy hozzájussanak a növekedéshez szükséges, újabb vállalkozói tőkéhez is. Az 1981 szeptemberében zárult pénzügyi évben már 70 dollárcentet fizettek részvényenként, forgalmuk pedig 335 millió dollár volt. Az üzembe helyezett Apple II rendszerek száma már fél évvel korábban meghaladta a kétszázezret. Azt sem lehetett pedig mondani, hogy vetélytársak nélküli piacon érték el ezt a sikert.

Commodore és Tandy előzmények

Az eredeti személyiszámítógép-gyártók két másik óriásának fejlődése alapvetően különbözik az Apple-étól. A Commodore cég piacra lépését leginkább a mikroszámítógép-ipart megalapító MITS történetéhez hasonlíthatnánk. Az eredetileg kanadai székhelyű vállalkozás a kalkulátorpiacon próbált az első időszakban meghatározó szerepet kivívni magának. Ennek során megtanulta, hogy milyen döntő jelentőségű a technológia. „1968-ban 1495 dollárért árultuk zsebszámológépeinket. Nyolc évvel később ugyanazt a kalkulátort már 9 dollár 95 centért kínáltuk" – emlékezett vissza 1983-ban a cég alapító vezéralakja és akkori elnöke, Jack Tramiel.

Tramielnek saját bőrén kellett tapasztalnia, hogy a piacot úttörőként felfuttató új vállalkozásokkal szemben milyen kíméletlen keménységgel lép fel a „big business". A félvezetőgyártás óriásaként ismert Texas Instruments a 70-es évek első felében elhatározta, hogy átveszi az uralmat a kalkulátorpiacon. Kiváló minőségű és az addigiaknál többet tudó termékeivel megkezdte a piac eláraszását. A hurok először még csak a kis halak nyaka körül szorult. Közéjük tartozott a MITS is.

A MITS 1971-ben beindított, fejlett kalkulátor-üzletága 1974-ben majdnem csődbe juttatta a céget. A piaci helyzet változása miatt hirtelenében háromszázezer dollár fedezetlen tartozása keletkezett. Új termék után kellett nézni, megszületett az Altair 8800, ami viszont elindította a korszakalkotó jelentőségű mikroszámítógépes lavinát.

A szorítást hamarosan a nagyobb halak is megérezték. A kalkulátor-chipeket gyártó MOS Technology a megrendelések csökkenését tapasztalta. A 6502 típusú mikroprocesszorra alapozott KIM–1, egykártyás mikroszámítógép sikere ugyan pótolta valamelyest a kiesést, de kapóra jött a Texasszal szemben a saját félvezető-gyártóbázisra kényszerült Commodore cégvásárlási ajánlata. A MOS Technology birtokában Tramiel is úgy látta, hogy új termék után kell néznie. Felkérte Chuck Peddle-t, a 6502 és a KIM–1 főkonstruktőrét, hogy tervezzen egy házi célokra is alkalmas, komplett mikroszámítógépet.

Ebben a kezdeti időszakban még szorosan összekapcsolódik a későbbi nagy hármas története. A leendő triumvirátus harmadik tagja, a Tandy/Radio Shack is felfigyelt a kibontakozó mikroszámítógépes lehetőségekre. A cég elektronikai kötődése 1963-ra nyúlik vissza, amikor a bőráru-szakmában, családi vállalkozásként működő Tandy felvásárolta az akkor még csak gyerekcipőben járó, hobbi elektronikai kereskedelem egyik cégét, a Radio Shacket (Rádió Kunyhó). A 70-es évek közepén már világméretű bolthálózattal rendelkezett, ahol a csereakkumulátoroktól kezdve, a különféle elektronikus kacatokon keresztül, az „építsd magad" elektronikai építőkészletekig, mindent meg lehetett vásárolni. Több kiskereskedelmi egysége volt, mint a gyorsétel-szakmában vezető McDonald's-nak.

A Tandyvel üzleti kapcsolatban álló Tramiel a Radio Shack első specifikációs igényeinek megfelelően terveztette meg a később PET-ként ismertté vált Commodore gépet. Chuck Peddle és főnöke az 1977 januárjában rendezett Fogyasztói Elektronikai Kiállításon mutatták be terméküket a Tandy/Radio Shacknek. A két cég azonban nem tudott megegyezni, így a Commodore önállóan hozta forgalomba gépét 1977 második felében.

1977 elején a Commodore-nak volt még egy kísérlete, hogy az akkor színre lépett Apple-t felvásárlás útján csatolja magához. A konkurrencia kizárására irányuló ezen kísérlet is kudarcba fulladt, így a késő tavasszal megrendezett Nyugati parti Számítógépes Vásáron egyidőben jelentették be a PET-et és az Apple II-t. Az utóbbi gép szállítása hamarosan meg is kezdődött.

A Tandy közben saját konstrukcióján dolgozott. Erre egyik vevője, Gordon French beszélte rá, aki mellesleg a HCC egyik alapítója is volt. French előzetes konstrukciós tervét a cég igen ígéretesnek találta. Az igazi fejlesztő mérnöki munkára Steve Leiningert alkalmazták, aki a National Semiconductornál szerzett tapasztalatok mellett a piacnak is jó ismerője volt, mivel a Byte Shop számítógépes üzletében is dolgozott munka után. A PET-tel gyakorlatilag egy időben piacra került TRS–80 azonnal sikert aratott. Az első hónapban 10 ezer darabot adtak el, háromszor annyit, mint az első évi teljes értékesítési terv volt.

Termékeny verseny

Az Apple-nek és a Commodorenak igencsak igyekeznie kellett ahhoz, hogy felvegye a versenyt a saját boltokkal rendelkező, nagy riválissal. Pénzügyileg sem kisebb hatalmassággal kellett szembenézniük, mint az 1978 júniusában befejeződött pénzügyi évet 1 milliárd dollár forgalommal záró óriással. A részletes forgalmi adatokat nem közlő vállalat eladásait 60 és 100 ezer darab közöttire becsülték az akkori külső elemzések.

A TRS–80 fő vonzereje volt, hogy az akkori idők legnagyobb teljesítményű mikroprocesszorára, a Zilog Z80-ra épülő 4 kbájtos alapgépet mindössze 599 dollárért kínálták. Billentyűzete az igényesebb alkalmazásoknak is megfelelt. Igényesebb megjelenítéshez a Tandy külön monitort kínált. Már az első évben a legteljesebb periféria- és szoftverválasztékot ajánlotta a többiekhez viszonyítva. Egy komplett rendszer ára persze ekkor már elérte a néhány ezer dollárt is. A vevők széles körből kerültek ki: kisvállalatok, újonc számítógép-felhasználók, oktatási intézmények és a kihelyezett célalkalmazásokat megvalósítani kívánó nagyvállalati részlegek. A fejlesztés, gyártás, értékesítés és vevőszolgálat egy kézben tartása viszonylagos biztonságot is adott az akkori vevőnek.

A PET jellemzője volt, hogy valamennyi lényeges elemet: egy 9 hüvelykes képernyőt, kazettás magnót és billentyűzetet szolgáltatta a 8 kbájt RAM kapacitású és 795 dolláros árával rendkívül olcsó alapkonfigurációjában. A bővíthetőségről a beépített IEEE–488 (HP–IB) interfész gondoskodott, úgyhogy a laboratóriumi és egyszerűbb folyamatirányítási alkalmazásokhoz is könnyen lehetett használni (műszerek csatlakoztatása). A Commodore-nak legfőbb tartaléka volt a többiekkel vívott versenyben évtizedes gyártási tapasztalata és saját félvezetőgyára. Ezzel a nagy tömegű és minimális költségű gyártásra kellett helyeznie középtávú stratégiájában a hangsúlyt.

Az Apple II gép volt valamennyi között a legkorszerűbb konstrukció. A 995 dolláros árért színes és nagy felbontású grafikát, beépített hangszórót, analóg bemeneteket, BASIC értelmező programot, valamint más ROM-ba égetett firmware-t, 48 kbájtos RAM bővíthetőséget és könnyen használható, perifériális illesztőkártya-helyeket kapott a vevő. A cég fő erőssége volt, hogy hardver és szoftver tekintetében egyaránt nyitott rendszerével támaszkodni tudott más hardver- és szoftvergyártókra is, az igen kisszámú, de mégis kommersz katalógusáramkör használata pedig lehetővé tette a gyártási önköltség versenyképes szinten tartását. Ezzel a többiek piacán is versenyképesnek bizonyult, sőt az egyetlen személyi számítógép volt, amit fejlesztő gépként is lehetett használni. Az új és ennek megfelelően legkorszerűbb gyártóbázis kiépítésének szükségszerűsége kompetitív előnyhöz juttatta az

Az árengedmény mértéke (a listaár %-ában) vevőnek → / eladótól	Hardvergyártó	Saját márkajelzésű OEM	Kiadó	Nagykereskedő	Kiskereskedelmi bolthálózat	Franchising központ	Kiskereskedő	Értéktöbblet rendszerház/OEM	Árengedményes üzlet/csomagküldő v.	Nagy tételben vásárló felhasználó
Hardvergyártó		50–60		35–50	35–50	35–50	10–40	5–35	35–75	5–35
Szoftvergyártó	50–85	50–85	50–85	40–60	40–60	40–60	35–50	35–50	40–85	5–50
Saját márkajelzésű OEM				35–50	35–50	35–50	10–40	5–35	35–75	5–35
Kiadó	50–85	50–85	50–85	40–60	40–60	40–60	35–50	35–50	40–85	5–50
Nagykereskedő					25–50	25–50	15–40	5–35	15–40	5–35
Kiskereskedelmi bolthálózat								5–15	50–?	5–35
Franchising központ							15–40	5–25	50–?	5–25
Kiskereskedő								5–25	50–?	5–25
Értéktöbblet rendszerház/OEM										5–25
Árengedményes üzlet/csomagküldő vállalat										10–75

Megjegyzés: Az OEM rövidítésnek az Other Equipment Manufacturer és az Original Equipment Manufacturer egyaránt megfelel. A lényegen ez semmit nem változtat, mivel arról van szó, hogy a vett terméket eredeti (tehát más) konstrukcióvá téve továbbadja a vevő, netán saját cég- és márkajelzéssel ellátva.

A kereskedelmi kapcsolatok rendszere és árviszonyainak alakulása az USA-ban (árengedmény a listaár százalékában)

Apple-t, hiszen a beruházásokat saját profitja mellett vállalkozói tőkével, illetve a későbbiekben részvénykibocsátással tudta finanszírozni.

A versenyből végül egyik cég sem került ki veszteségként, mivel a személyiszámítógép-gyártók e sajátos kombinációja egy korlátlan felvevőképességű piaccal találta szembe magát. Mi több, a verseny – talán az egyetlen Tandy kivételével – kellően megedzette őket, hogy a hamarosan piacra lépő „big business"-szel is bátran szembe tudjanak nézni.

Mindennek haszonélvezői maguk a vevők voltak, akik elérhető áron, szinte házhoz szállítva juthattak hozzá az új személyi számítógépes technikához. Választási lehetőségük lévén, hathatósan tudták befolyásolni a kínálat alakulását, és így belső törvényei szerint tovább növekedhetett a mikroszámítógépes lavina.

Hatékony piaci rendszer

Az Altair-korszak mikroszámítógépes háziiparával még csak éppen hogy fejlődésnek indult szaküzletrendszer a Commodore és az Apple piacra lépésével újabb, jelentős ösztönzést kapott. Egyre bővült a kedvezményes árakkal vonzerőt teremtő postai csomagküldő (mail order) vállalkozások forgalma is. Amíg a számítógépes szaküzletek helyi alapokról indulva terjesztették ki tevékenységüket földrajzi és szakterületi irányokban, addig a csomagküldő kereskedelem rögtön országos méretekben kezdte meg tevékenységét. Az utóbbiaktól viszont nem lehetett semmiféle szolgáltatásra számítani a kínált termékekkel kapcsolatban.

Kezdetben két-, sőt háromszintű volt a kereskedelmi rendszer. A gyártóhoz kötődő nagykereskedelemben az induláshoz szükség volt olyan cégekre, amelyek regionálisan és azon belül esetleg szakterületileg is képviselni tudták a gyártót. A Commodore és az Apple is ily módon építették fel számítógépes kereskedelmi hálózatukat. Amint megjelentek a gyártófüggetlen, közvetítő nagykereskedők (distributors), a képviseleti rendszer kezdett veszíteni jelentőségéből. A képviselő ugyanis csak a megrendeléseket közvetítette, a nagykereskedő pedig saját készletére, maga vásárolt.

A kereskedelemben is egyre fokozódó verseny a 80-as évek elején már a kétszintű kereskedelmet is kezdte megingatni. A tisztán közvetítő nagykereskedőket kiiktató kereskedelmi rendszerek tudták ugyanis leginkább bővíteni forgalmukat az elmúlt években. Ennek egyik úttörője volt a Computer-Land. 1977-ben nyitotta első üzletét Kaliforniában. Piaci terjeszkedésének alapfilozófiája volt az ún. franchising. A hálózathoz tartozó önálló kereskedelmi vállalkozások igényes üzletviteli know-how-t, egységes megjelenést és országos publicitást kaptak. A Computer-Land központ a gyártókkal szemben nagykereskedőként lépett fel, és szolgáltatásaiért nagyon kedvező árat számolt fel. Az állandó tagsági díj mellett mindössze 3–8 százalék közötti, forgalom utáni jutalékot kellett fizetni a kiskereskedelmi vállalkozásoknak. 1983-ban már 25 ország 620 üzlete tartozott a ComputerLand-lánchoz. Az éves forgalom elérte a 963 millió dollárt. Az üzleti csoportosulás pénzügyi stabilitása és vevőszolgálati rendszerének igényes kialakítása mind a gyártók, mind a vevők szemében egyre növelte a Computer-Land kereskedelmi jelentőségét.

CP/M feltámadás PC köntösben

A piaci rendszer fejlődése lehetővé tette, hogy az eredeti Altair-IMSAI gyökereken fogant CP/M vonal kiheverje az 1980-as év hirtelen támadt értékesítési nehézségeit. A MicroSoft elkészítette az Apple II-t CP/M géppé változtató SoftCard-ot, így saját fordítóprogramtermékei mellett az akkori idők legteljesebb szoftverválasztékának is új csatornát nyitott. Az 1978-ban alapított MicroPro szófeldolgozó programcsomagja, a WordStar is növelhette eladásait. (Napjainkig 750 ezer példányban adták el.) A hatékony piaci rendszer és a CP/M szoftverbázis új személyiszámítógép-gyártók színre lépését is lehetővé tették.

A CP/M piacon támadt átmeneti nehézségeket kitűnő érzékkel használta ki Adam Osborne, aki a mikro-számítástechnika könyvkiadási piacán szerzett korábban hírnevet magának. Mindössze 55 ezer dollárért korlátlan értékesítési licencet szerzett a CP/M-re, és hasonlóan előnyös feltételek mellett jutott hozzá a CBasic és MBasic rendszerekhez, valamint a Word-Starhoz is. Kedvező fejlesztési szerződést kötött a Sorcimmel egy VisiCalc-szerű CP/M alkalmazási program, a SuperCalc fejlesztésére is.

Az 1981 elején lezárult megállapodások eredményeképpen kiderült, hogy nagy sorozatban gyártandó CP/M gépéhez a géppel együtt adandó szoftverkészlet ára nem lesz több, mint gépenként 10 dollár. Időközben az új gép prototípusa is elkészült (Lee Felsenstein, HCC tag és régi konstruktőr tervezte). Fő attrakciója volt 64 kbájtos, bőséges RAM kapacitása, 2×90 kbájtos floppykapacitása, beépített monitora, billentyűzete és interfészei mellett, hogy mindezt szállítható dobozolásban és szuperalacsony áron, mindössze 1795 dollárért kínálta.

Az Osborne–1 1981. júliusi megjelenésével a legkorábbi indíttatású mikrogépes hullám piacilag is érett szakaszához érkezett. Az Osborne-i utat kisebb-nagyobb eltérésekkel mások is követték. A CP/M személyi számítógépek így méltó versenytársai lehettek az elkövetkező három évben a gyártóspecifikus rendszereknek, és nem szorultak a pótkártyás megoldások perifériájára. A mikroszámítógépes lavina igazi törvényszerűségeit azonban a számítástechnikai ipar marketing királya, az IBM ismerte fel.

NACSA SÁNDOR

INFORMÁCIÓTECHNIKAI VÁLLALAT

ORSZÁGOS SZERVIZHÁLÓZATUNK

Központ: Budapest V., Bécsi u. 8.
Levélcím: Budapest, Postafiók 314. 1369
Telefon: 22-4381 22-6841

korszerűen felkészített műhelyei útján végezzük a számítástechnikai berendezések műszeres vizsgálatát, bemérését, ellenőrzését.

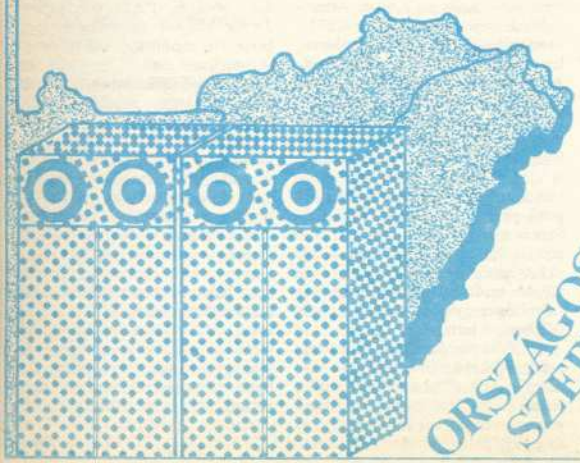
VÁLLALJUK:

adatrögzítő- és feldolgozó berendezések,
személyi számítógépek,
kisszámítógépek,
számítógép-rendszerek
garanciális és garanciaidőn túli javítását, szervizkiszolgálását.

KERESKEDELMI főosztályunk Budapest III., Kerék u. 6. útján FORGALMAZUNK

- robotron A 6402 típusú számítógépeket,
- TZ 80 típ. professzionális számítógéppel vezérelt munkaidő-rögzítő és nyilvántartó rendszereket,
- különféle személyi számítógépeket,
- különböző típusú számítógép printereket,
- számítógépek mágneses perifériáihoz szükséges adathordozókat:

  floppy diszkeket, mágnesszalagokat, mágneslemezeket,
- számítástechnikai berendezésekhez és eszközökhöz szükséges alkatrészeket



SZOLGÁLTATÁSAINKKAL ÉS ÁRUKINÁLATUNKKAL TISZTELT VEVŐINK RENDELKEZÉSÉRE ÁLLUNK!

SW.'86

data manager

Ma már a kis gépek is nagy teljesítményre képesek
... ha Ön körültekintően vásárol, gondosan tervez és időben fejleszt, amikor a számítógépre bízza feladatait.

SEGÍTHET, HA SEGÍTHETÜNK!

Mi nem csak árusítjuk, leasing-eljük az IBM és SIRIUS (VICTOR) számítógépeket, hanem előkészítjük, megtervezzük, üzembe helyezzük és karbantartjuk az Önök feladatához szükséges rendszert. Tapasztalt hardver- és szoftverfejlesztők, szervezők és programozók szolgálják ki Önt, hogy a legmegfelelőbb géppel és programokkal magas színvonalon dolgozhasson.

Mi nem csak egyszeri kapcsolatra törekszünk. IBM PC-k és más, kompatibilis számítógépek alkalmazóinak szerveztük az IBM KLUB-ot. Az eredményes gazdálkodást támogató információkkal, szolgáltatásokkal, kedvezményekkel várjuk a felhasználókat.

Mi nem csak az IMB-es felhasználókkal törődünk. Kis- és mikrogépes partnereink jobb kiszolgálása érdekében az ország különböző pontjain működő GMK-kal olyan hálózatot építettünk ki, amely a DATA MANAGER irányításával lehetővé teszi a referenciával rendelkező, esetleg csak speciális, szűk területen tevékenykedő vállalkozók felkutatását, szolgáltatásaik és termékeik megismerését, megismertetését. Ez a hálózat biztosítja, hogy az Önök által forgalmazásra felajánlott piacképes terméket rövid időn belül az egész országban ajánljuk, bemutassuk az érdeklődőknek.

SEGÍTHET, HA SEGÍTHETÜNK!

1985. november 12. és 15. között keressen fel a **Hotel Duna Inter*Continental**-ban a *SOFTWARE'86* kiállításon. Széles körű szolgáltatással, új fejlesztésekkel várjuk Önt.

A Commodore 64 kommunikációs lehetőségei

A mikrogépek elterjedése a számítástechnikában nagy fejlődés kezdetét jelentette. Felvetődik azonban néhány olyan kérdés, amely bosszúságot okozhat azoknak, akik több számítógépen dolgozhatnak. Ezek közül az egyik leglényegesebb a kompatibilitás problémája, tehát az, hogy az egyik fajta gépen kifejlesztett program vagy felvett adatállomány a másikon is futtatható legyen, illetve a másik gép is használni tudja.

Magyarországon ez a probléma még súlyosabb, mert a különböző csatornákon teljesen véletlenszerűen bekerült gépek sokfélesége és az árak meghatározó szerepe miatt szinte nincs is mód arra, hogy a gépek megvételekor olyasmire is gondolhassunk, hogy majd később, ha lesz egyáltalán lehetőség rá, igényeink növekedtével a kidolgozott programjainkat más, „nagyobb tudású" gépekre is átvihessük. Pedig aki már megpróbálta, az tudja, hogy nem olyan élvezetes dolog egyszer már megírt programjainkat újra forrásnyelven bevinni egy másik gépen csak azért, mert a két gép nem képes olvasni például ugyanazt a lemezformátumot.

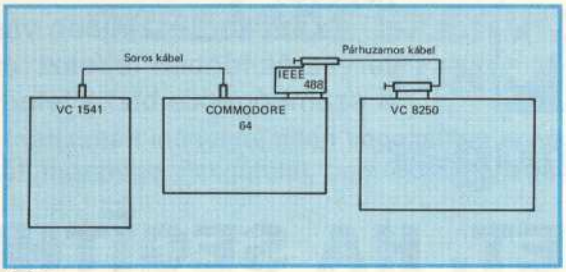
A személyi számítógépek nyelve a BASIC. Szintaktikájuk is hasonló, nem kellene tehát sokat változtatni a programokon ahhoz, hogy használni tudjuk máshol is. De hogy vigyük át a programokat egyik gépről a másikra?

Miből adódnak ezek a kompatibilitásbeli problémák? Egyszerű lenne a helyzet, ha ugyanazt a floppyt használhatnánk minden géptípusra. Maguk a lemezek többé-kevésbé alkalmasak lennének erre, de a formázást, lemezkezelést a különböző lemezmeghajtók különbözőképpen oldják meg. (A Commodore 64 VC 1541 típusú meghajtója például 172 kbájtot, ugyanakkor a Commodore 720 VC 8250 meghajtója kb. 1 Mbájtnyi információt képes tárolni ugyanazon a lemezen.) Természetesen a két gép nem tudja egymás lemezeit olvasni.

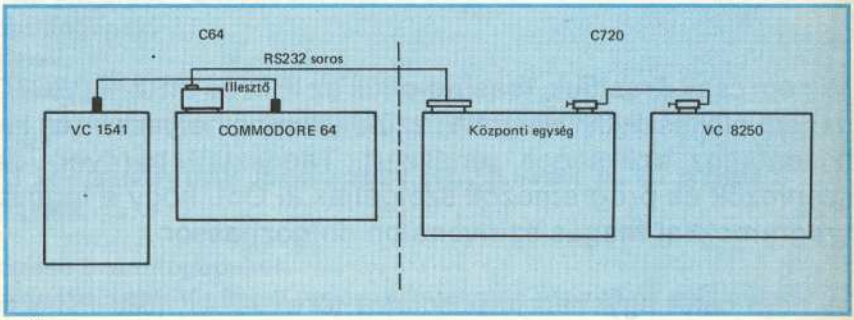
A továbbiakban e két gép közötti átvitelre teszünk javaslatokat, de utalunk a C64 egyéb gépekhez történő kapcsolására is.

Többféle megoldás is van, amelyeket elsősorban hardverlehetőségeink határoznak meg.

1. A C64 bővítő (expansion) portjához lehet olyan kiegészítő panelt kapni, amelynek segítségével az IEEE-488 párhuzamos csatlakozón keresztül a VC 8250 nagy lemezmeghajtót közvetlenül csatlakoztathatjuk a C64-hez *(1. ábra)*. Ilyenkor azonban a C64



1. ábra



2. ábra

3. ábra

nem képes kezelni a perifériákat a 4–11 csatornákon, csak a nagy lemezt. Közvetlen paranccsal azonban:

POKE 57221,PEEK(57221)
OR 2↑(DN−4)

(ahol DN az érintett készülék címe), fel lehet szabadítani a régi perifériák közül a szükségeseket.

Például 8-as lemezmeghajtó esetén:

POKE 57221,PEEK(57221)
OR 2↑4

visszaállítás:

POKE 57221,PEEK(57221)
AND (255−2↑(DN−4)).

A bővítő lehetőséget nyújt arra is, hogy a BASIC 4.0 egyszerű lemezkezelő utasításait használhassuk. Alkalmazzuk a SYS 36864 utasítást!

Az átvitel a következőképpen történik:

– Kapcsoljuk össze az 1. ábrának megfelelően a gépeket!

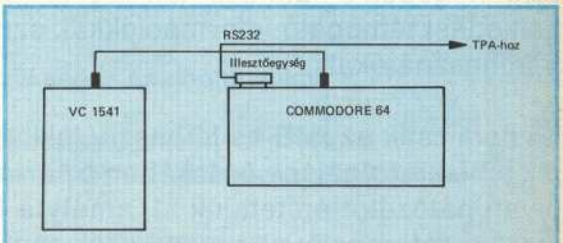
– Kapcsoljuk be a központi egységet és a VC 1541-es lemezmeghajtót, de ne kapcsoljuk be a VC 8250-es meghajtóegységet!

– Szabadítsuk fel a POKE utasítással a 8. egységet.

– Töltsük be a szokásos módon a lemezegység-átcímező programot, majd állítsuk vissza az újabb POKE utasítással az eredeti helyzetet.

2. A másik megoldás egyszerűbb lenne, ha létezne igazi RS232 soros kimenet a C64-ben. A kivezetések megjelennek a felhasználói ponton, de TTL szinten, holott a normál RS232 ±12 volttal dolgozik. Így megfelelő szintillesztésre van szükség. Létezik egyébként eredeti Commodore RS232 interfész. Ennek segítségével csatlakozhatnánk olyan más számítógépekhez, amelyek szintén rendelkeznek soros RS232 kimenettel.

Ebben az esetben a kommunikáló programokat megírhatjuk BASIC nyelven vagy gépi kódban.



A BASIC nyelven írt programok használják az OPEN, CLOSE, CMD, PRINT#, INPUT#, GET# utasításokat. Például:

OPEN 1,2,Ø,CHR$(6) – megnyitás 300 Baudos sebességre.

Az adatátadást 2 db 256 bájtos puffer segíti, amelyek a memória tetején helyezkednek el. Ezeket az OPEN utasítás aktivizálja, a CLOSE zárja.

Csak azoknak, akik értenek a hardverhez! A legegyszerűbb a 3 vonalas interfész, amelyen 3 vonalon keresztül történik megfelelő szintillesztéssel az átvitel.

A felhasználói port
B – S_{in} – bemenet
A felhasználói port
M S_{out} – kimenet

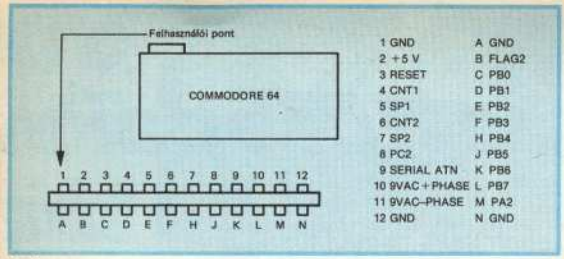
– Kapcsoljuk ki a VC 1541-es meghajtót, kapcsoljuk be a VC 8250-est, és címezzük át 8-ról 9-es címűre, majd ismét szabadítsuk fel a 8. egységet.

– Most már bekapcsolhatjuk a kis lemezt is.

– Adjuk ki a SYS 36864 utasítást.

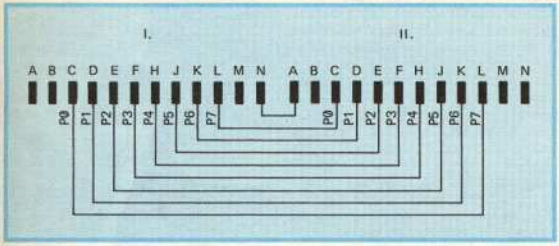
Ezután a szokásos módon betöltve a programokat és kimentve a másik lemezre, az átvitel könnyen megoldható.

A nagy lemezt újra a Commodore 720-hoz kapcsolva elolvashatjuk az átvitt programokat. Természetesen a szintaktikai hibákat ki kell javítani. A képernyőkezeléssel is baj lesz, mégis jelentős megtakarítást értünk el, hiszen csak javítani kell, és nem beírni az egész programot.



4. ábra

5. ábra



A felhasználói port
A,N – föld
A felhasználói port
C,D,E – magasban (feszültség alatt)
Az elvi megoldás a *2. ábrán* látható.

Az átvivő programnak a megnyitás után tartalmaznia kell egy kódkonvertáló programrészletet, mert a C64 más kódot használ, mint a szokásos ASCII. Tehát a Commodore kódjait ASCII-ba kell konvertálni, illetve a fogadottakat vissza. (A táblázat kézikönyvekben megtalálható.)

Ezt a konvertáló részt használhatjuk lemezről való átvitelnél is, csak előbb egy kis trükköt kell alkalmazni. A lemezre is ilyen kódban kell előbb lerakni a programot. Ez a következőképpen történik. Beolvastatjuk a programot a központi egységbe, majd

```
OPEN 2,8,2,"FILENÉV,U"
CMD2
LIST, majd a kimentés után
CLOSE 2
```

Tehát a képernyő helyett a lemezre listáztattuk a programot, amely így már a konvertáló program segítségével átküldhető a másik gépre.

A programokat természetesen ebben az esetben is javítani kell, hiszen itt is csak az átvitelt valósítottuk meg, a gépre illesztést nem.

Ugyanez a megoldás, ha a C64-et nagygéphez, például TPA-hoz illesztjük *(3. ábra)*. Ilyenkor a C64 mint intelligens terminál kapcsolódik a TPA-hoz.

3. A harmadik lehetőség a felhasználói port használata a kommunikálásra, közvetlen utasításokkal (természetesen TTL szinten), például két C64 összekötésére *(4. ábra)*. Ez már nem tartozik ugyanabba a csoportba, amiről eddig beszéltünk, hiszen két C64 egymással kompatibilis; de szükség lehet rá, például hogy két gépen két játékos tudjon egymással kommunikálni. Ez a megoldás korlátozottabb az eddigieknél.

Az adatok átvitele a PB0–PB7 vonalakon valósítható meg. Hogy ezek közül melyik legyen kimenet és melyik bemenet, azt az AdatIrányRegiszter (AIR) beállításával szabhatjuk meg.

Az AdatIrányRegiszter
A porthoz 56578 címen,
B porthoz 56579 címen található.
Maguk a portok
A port 56576 címen,
B port 56577 címen találhatók.

Az A, illetve B portnak megfelelő beállítások csak mindkét cím együttes beállításával történhetnek.

Állítsuk be például a B portot úgy, hogy a
PB0–PB1–PB2–PB3 – kimenet
PB4–PB5–PB6–PB7 – bemenet legyen.

POKE 56579,15 mert

BIT#:	7	6	5	4	3	2	1	0
ÉRTÉK:	0	0	0	0	1	1	1	1

15

A bevitel a PEEK(56577), a kivitel a POKE 56577, output utasítás segítségével történik. Ennek a példának megfelelő összekötés látható az *5. ábrán.*

A bemutatott lehetőségek természetesen csupán hiánypótlási célzattal születtek, és nem térnek ki minden részletre. Aki azonban rá van utalva, hogy hasonló dolgokkal foglalkozzon, talán könnyebben elindul ezek után.

VARJU ATTILÁNÉ–SZÍVÓS GÉZA

Ismerkedjen meg legújabb termékeinkkel!

MÁTKA DADEX VETERV

TAMARA GAIA

IBM PC

SZÁMALK

A szakembergárdája készségesen áll minden kedves ügyfele rendelkezésére

SW.'86

Minden kedves érdeklődőt várunk a Hotel Duna Inter*Continental BUDAVÁR Terem 35. standján 1985. XI. 12–15. között

AZ ORSZÁG MINDEN RÉGIÓJÁBAN EGSZI

SPECIÁLIS – JOGVÉDETT – ALAPSZOFTVER JELLEGŰ RENDSZEREK

Mg-80 – Fejlesztő és üzemeltető rendszer párbeszédes adatfeldolgozáshoz

ERPEL – Egységes rendszer és programfejlesztési eljárás

KÖNYVVITELI PROGRAMOK IBM PC–XT-RE

MIFOX – Mikroszámítógépes főkönyvi könyvelési rendszer

MIFOLK – Mikroszámítógépes folyószámla könyvelési rendszer

ÜGYFELEINK KEDVEZŐ FELTÉTELEK MELLETT VÁSÁROLHATJÁK MEG, VAGY VEHETIK BÉRBE PROGRAMCSOMAGJAINKAT!

LEGÚJABB RENDSZEREINK TAP–34 ÉS MO8X MIKROSZÁMÍTÓGÉPEKRE

HELP – Keretrendszer mikroszámítógépre
KERSZAB – Kereseti adót számító és bértervezést segítő programcsomag
REGULA – Vállalati gazdasági és keresetformatervező rendszer
ERTERV – Építőipari vállalatok számítógépes eredménytervezési rendszere
RMKF – Vásárolt anyagok, fogyóeszközök központi raktári és munkahelyi készletének, forgalmának kimutatása programcsomag
SZTH – Szerződésteljesítés és számlázás programcsomag
TFTK – Házgyári és építéshelyszíni termelést követő programcsomag
PIR – Mikroszámítógépes pénzügyi információs rendszer
MIFOK – Mikrogépes főkönyvi-könyvelési rendszer
MIFOLK – Mikrogépes folyószámla-könyvelési rendszer
MUNYIR – Mikroszámítógépes munkaerő nyilvántartási rendszer
RUSZER – Mikroszámítógépes munkaruha- és szerszámnyilvántartó programcsomag
DMGR – Demográfiai tervezést, elemzést segítő és előrejelző információs rendszer
UTVIL – Tér- és útvilágítástervező interaktív program
KÖLTS – Építmények tervezői költségbecslés készítését támogató számítógépes programcsomag MO8X prof. személyi számítógépre

SW.'86

INTÉZETÜNKTŐL
IBM PC–XT
MO8X ÉS TAP–34 MIKROSZÁMÍTÓGÉPEKET BÉRELHET!

ÉPÍTÉSGAZDASÁGI ÉS SZERVEZÉSI INTÉZET
1027 BUDAPEST
II., CSALOGÁNY U. 9–11.
TELEFON: 153-660

AZ ORSZÁG MINDEN RÉGIÓJÁBAN EGSZI

az európai v
he

Nem sportverseny ez, nem valami aranyéremért küzdenek a nyugatnémet gyártók, a nagy világcégek NSZK-beli képviselői, az importőrök, a kereskedők. De iparuk fejlettségével nem tartják összeegyeztethetőnek, hogy Franciaország és Anglia megelőzi őket a számítástechnika alkalmazásában. Cégeik kevesebbet költenek számítógépekre, szoftverekre, perifériális berendezésekre, mint a franciák és angolok; még nem tudták behozni azt a lépéshátrányt, amelybe azért kerültek, mert a másik két ország, amely a második világháború győztes hatalmai közé tartozott, hamarabb fogott a számítástechnika fejlesztéséhez.

Hogy a számítástechnika alkalmazása az élet minden területén milyen rohamosan fejlődik: közhely. Nem kevésbé közhely az, hogy a felhasználás szélesítése kihat a termelésre, az eszközök forgalmára. S mivel a gyártók, a kereskedők forgalmuk növelésére törekszenek, és vannak is elképzeléseik a felhasználási területek bővítésére, kézenfekvő a tennivalójuk: tudatosítani a kiaknázatlan lehetőségeket.

Ezt tapasztaltam a közelmúltban is, a kölni Computer Show '85 kiállításon. Másodízben rendezték meg az idén – és a terv szerint ezután minden esztendőben – ezt a nagyközönségnek szóló, impozáns bemutatót. Egy vásár mindig az információk terjesztésére szolgáló médiák koncentrált felhasználását teszi lehetővé és igényli is; ez alól a kölni sem kivétel. Tapasztalataim szerint három média bizonyult különösen hatásosnak. Ezekkel foglalkozom a következőkben.

Könyv a termékkel egyidőben

Az egyik feltűnő jelenség volt a számítástechnikai szakkönyveket kiadó vállalatok nagy száma. Érthető, hiszen ez a milliókat megmozgató, viszonylag új szakma részletekbe menő, széles és nagyon szakszerű tájékoztatást igényel. Az veszi meg a gépet, az használja a kínált programokat, aki pontosan tudja, mire jók és hogyan kell használni őket. Az évi több száz könyvet forgalmazó nagy kiadók közül is mind többen állnak rá a számítástechnikai ismeretterjesztésre, és sorra alakulnak szakkiadók, amelyeknek egy része terméket – legalábbis szoftvert – is forgalmaz.

„Hidak az új tudományhoz" – hirdeti a düsseldorfi SYBEX, amely részben szoftverekhez kapcsolt nyolc termékismertető kiadványában 32 géptípusról informál, igen sokoldalúan; az Apple, Atari és Commodore gyártmányairól pedig 28 új könyvet ajánl. A FALKEN Verlag, a „hasznos könyvek specialistája", kilenc újonnan megjelent művet hirdet. A müncheni Franzis' kiadó –

amely öt folyóiratot is gondoz – ez évi katalógusában több mint 50 számítástechnikai szakkönyvet, köztük tíz újdonságot ajánl. A Texas Instruments NSZK-beli önálló cége tájékoztató szolgálatot tart fenn; kiadványában mintegy 25 könyvről olvashatunk, amelyek mind az amerikai cég termékeinek használatát könnyítik meg. Mindezek persze csak szemelvények, hozzávetőleg 40 kiadó termékéből.

Különösen felkeltették érdeklődésemet a düsseldorfi *Data Becker* kiadványai. Ez a cég a szoftverházak között szerepelt, de egységes tipográfiájú, jellegzetes narancssárga könyvei, amelyekből mintegy százat láthattunk, kie-

Az angol Apricot gépen grafikai versenyt rendeztek. A győztes ez az Effel-szerű karikatúra lett

A kiállítás nagysikerű eseménye volt Anatolij Karpov fellépése a sakknapon

melkedtek a kínálatból. A Commodore, Schneider, Apple, Atari, Sinclair, IBM PC gépek lehetőségeiről és a különféle szoftvertermékekről szóló munkáknak első pillantásra is a frissesége tűnt fel; olyan gépről – a Commodore 128-ról – is láttam könyvet, amely épp hogy a piacra került.

Heinz-Dieter Stops urat, az értékesítési osztály vezetőjét kérdeztem üzletpolitikájukról, módszereikről. „Legfőbb szempontunk" – mondta – „a közönség részéről várható kérdésekre szinte a készülék megjelenésével egyidejűleg válaszolni. A cégek szakembereitől szerzőink jó előrejelzéseket kapnak, amelyekből tapasztalt gárdánk kitalálja, milyen problémák adódhatnak. Tekintve a gyors informálás rendkívüli fontosságát a szakmában, két hónapnál hosszabb átfutási időt nem engedhetünk meg magunknak. A szerzők szinte párhuzamosan dolgoznak a szerkesztővel és a nyomdával. Volt rá eset, hogy a megírással együtt négy hét alatt készen lett egy könyvünk. Az Atari legújabb készülékéről szólt, amelyet a Hannoveri Vásáron mutattak be, és már ott volt a könyv is. Végig éjszakai műszakban dolgoztunk rajta. Egyébként 120 könyv- és programírónk van, ezen túlmenően 80 állandó és mintegy 100 szabadúszó munkatársunk. Igen erős a fejlesztő részlegünk, amelynek szakemberei már a készülő új gép paramétereiből tudják, milyen tippeket adjanak a majdani programozóknak, hogy eredményesen dolgozhassanak a géppel."

Színes, ötletgazdag folyóiratok

Mindenki tudja, hogy a nyugati országokban se szeri se száma a számítógépes bulvárfolyóiratoknak; sok pavilonban több van belőlük, mint a szexlapokból. A népszerű lapok is fontos szerepet töltenek be, hiszen főképp a fiatalság buzgó olvasójuk, de azt tapasztaltam: a számítógépek mind nagyobb teret kapnak a komoly műszaki és gazdasági, sőt a napi sajtóban is.

A *Kölnischer Rundschau* pavilonjában bemutatta a szinte teljes egészében számítógépesített szerkesztési munkát.

Nagy tábla hirdette: „Az újságot gyorsabban írják, mint olvassák!" – s ez valóban így van. A vásár tartama alatt a lap a közönség szeme láttára készítette különszámait, amelyek nemcsak az eseményekről tájékoztattak, de valóságos műszaki és közgazdasági keresztmetszetet adtak a számítástechnika alkalmazásáról az NSZK-ban. Néhány kiragadott példa: a kereskedő naprakészen tájékozódhat a piaci kínálatról és az árakról; a játék forradalma; a kölni tanítók nyilatkozata a számítógépes oktatás formáiról; „házasság" a postával a BTX (képernyő-szöveg) sokoldalú használatában; a számítógép kiszorítja az írógépet; az építő robot; a diszkketteket gondozni kell stb.

A nyugati világ egyik legtekintélyesebb gazdasági lapjának, a Handelsblattnak heti magazin-melléklete teljes egészében a számítástechnikával foglalkozott, páratlan sokoldalúsággal. Palettáján nemcsak számos szellemes és szép, színes illusztráció, de olyan „színfoltok" is szerepeltek, mint „Egyes hardverek sokat tudnak, mások még többet" (az óriási műveleti teljesítményű chipekről); „A forró piac" (a számítógép minden egyes vevő megnyeréséért harcol); „Kemény pénzért puha áru" (a programok az ürszondák irányításától a sarki hentes vevőkörének biztosításáig terjedő, széles alkalmazási területet fednek le); „Az egyformán gondolkodók" (a digitális számítógép 'lelki élete': az aprócska chipek mind több funkciót látnak el); „A biztonság" (a visszaélések megakadályozásáról); „Karambolok" (amikor valami nem sikerül a számítógépnek) stb. Íme komoly, a műszakiak számára is olyan nehezen megérthető szakmánk tálalása – magazin módra.

Közönségkapcsolatok

A korszerű médiák egyik legfontosabbika, a PR (Public Relations), amely lényegében az érdekeltekkel való kapcsolatok ápolását jelenti, mind nagyobb szerephez jut a számítástechnikai ismeretek terjesztésében. Ez volt számomra a kölni Computer Shownak talán legérdekesebb, mindenképpen igen elgondolkoztató tapasztalata.

„Másképpen, újszerű eszközökkel kell a számítástechnikát demisztifikálni" – mondta a két csinos fiatalasszony igazgatta *Henschel+Stinnes* müncheni PR-iroda egyik cégfőnöke, Frau *Katherina Stinnes*. Bizonyára szebben is fordíthattam volna a fenti mondatot, de németül így mondta: „entmistifizieren", s ha azt mondom, megfosztani misztikumától, nem vagyok pontos. Valóban: a számítógép sokak számára valami megfoghatatlan csoda, de tekintsük ezt a csodát épp olyan mindennapinak, természetesnek, mint amilyen az emberi szervezet vagy az ökológiai egyensúly, vagy – ma már annak látjuk – a csillagos égbolt. Megérteni nem kell mindenkinek, de együtt élni vele – igen.

„Cégünk kizárólag a számítástechnikával foglalkozó vállalatok közönségkapcsolatainak intézését vállalja" – mondta Stinnes asszony. „Az NSZK-ban négy vásár van, amelyen számítástechnikai termékeket mutatnak be: a hannoveri nagy vásár, a müncheni System, a kölni Orgtechnik és ugyanitt tavaly óta a Computer Show. Ez az egyetlen, amely kizárólag számítástechnikát, és azon belül is mikrogépeket, és ezekhez való szoftvereket, tartozékokat mutat be. Mindegyiken speciális rendezvénysorozattal jelentkezünk. Hannoverben ilyen volt A fiatalság és a technika, itt pedig ilyen az Aktion-Center. Ezek a mi ötleteink, és hasonlókat igyekszünk a jövőben is megvalósítani. Mi a nagyközönséghez szólunk, és az elmúlt két évben eljutottunk oda, hogy a számítógépek darabszámát tekintve megelőztük Angliát, közel vagyunk a franciákhoz. Az iskolai számítógépes rendszer is jó irányban fejlődik, integrált hálózatokat alakítanak ki, amelyekben egy-egy nagyobb iskolai géphez 200–250 számítógépes munkahely csatlakozik. Ez óriási mértékben fokozza az alkalmazási lehetőséget. Nagy akciókat szervezünk az iskolai alkalmazás bővítéséért, amit sok esetben társadalmilag finanszíroznak."

Az Aktion-Center „talk-show"-inak sorában az egyik program címe Az asszony és a számítógép volt. A hozzászólók többsége megegyezett abban, hogy nem volt még technikai eszköz, amelyhez az asszonyok olyan közel kerültek volna, mint a számítógéphez. A talk-show-k akkora közönséget vonzottak, amelyhez hasonlót ilyen rendezvényeken még nem láttam. Nem csoda, a program valóban rendkívül érdekes volt. Szerepelt például a szoftverlopás problémája, amelynek vitáján kifejtették, hogy a szoftvert, mint szellemi tulajdont védeni kell, és ha valaki sikerrel lemásol egy védett szoftvert és ezért feljelentik, a bíróságon úgy kezelik, mint valami bűnözőt, és a becsületén esett foltot nem mossa le többé semmi, akkor sem, ha a vádlottat felmentik.

Komoly szakmai vita alakult ki a 16 bites processzor körül, amely abban csúcsosodott ki, hogy meg kell várni, míg ez is olyan olcsó lesz, mint a jelenlegi 8 bites, és akkor az alkalmazási terület határa a csillagos ég...

Az Aktion-Center szenzációja azonban a sakknap volt, „Sakk a világbajnokoknak", két programponttal. Elsőként a Kölnischer Rundschau által kiválogatott 30 amatőr sakkozó 30 tavalyi mikroszámítógép-világbajnok Mephisto ellen sakkozott. A jórészt amatőrökből álló társaságnak összesen hét tagja nyert az ördögi masina ellen. Ezt követően *Anatolij Karpov* sakkvilágbajnok lépett az Aktion-Center emelvényére, hogy húszjátszmás szimultánjátékkal szórakoztassa ellenfeleit és az érdeklődőket. Az egyik táblán a müncheni Hegener+Glaser cég Mephistoinak legújabb – úgy tűnik, a tavalyi világbajnoknál is jobban sakkozó – típusa, bizonyos B+P jelű volt az ellenfele. Ezt a játszmát kivetítették képernyőre, miközben *dr. Helmut Pfleger* nagymester, az NSZK egyik legkiválóbb sakkozója kommentálta a lépéseket. S a számítógép kis híján legyőzte a világbajnokot! Nyerő állása volt, de a végén már ő maradt az egyetlen ellenfél, túl gyorsan kellett lépegetnie, és ha nincs ideje számolni, az minden sakkszámítógép vesztét okozza.

A számítástechnika azonban nagy győztese volt ennek a nem mindennapi mutatványnak is.

LINDNER LÁSZLÓ

ZX MICRODRIVE 1

Programok felvétele

A ZX-Spectrum tulajdonosok többsége kezdetben kazettás magnetofont használt háttértárolóként. A megvásárolható gyári programok is zömében így voltak hozzáférhetők.

Az év elején a Sinclair cég megjelent a piacon egy, a tőle már megszokott koncepció alapján készült új háttértárral, a hozzá tartozó interfésszel és mikrokazettával. A MICRODRIVE alkalmazásával a ZX-Spectrum gépek igen hatékony és gyors adatmozgatást tesznek lehetővé, még a nagyobb tárigényű programok is néhány másodperc alatt a gépbe tölthetők.

Az új háttértárral bővített rendszert használók körében azonban gyakran felvetődik a kérdés: hogyan lehet a már meglévő programokat a MICRODRIVE kazettájára másolni?

Nehéz általánosan használható módszert megfogalmazni, hiszen a kapható gyári rendszer- és játékprogramok szervezése igen változatos. De az itt ismertetett programok, módszerek az esetek többségében eredményhez vezetnek.

A mellékelt program a magnókazettán levő program fejlécét olvassa ki. Ez olyan információkat tartalmaz a programról, amelyeket a másoláshoz feltétlenül ismernünk kell.

```
0000 F3          DI
0001 111100      LD DE,0011H
0004 AF          XOR A
0005 37          SCF
0006 DD21005B    LD IX,5B00H
000A CD5605      CALL 0556H
000D 30F2        JR NC,0001H
000F FB          EI
0010 DD21005B    LD IX,5B00H
0014 CD6B0D      CALL 0D6BH
0017 3E02        LD A,02H
0019 CD0116      CALL 1601H
001C 3A005B      LD A,(5B00H)
001F 11C009      LD DE,09C0H
0022 CD0A0C      CALL 0C0AH
0025 060A        LD B,0AH
0027 11005B      LD DE,5B00H
002A 13          INC DE
002B 1A          LD A,(DE)
002C D7          RST 10H
002D 10FB        DJNZ 002AH
002F CD7C00      CALL 007CH
0032 3B          DEC SP
0033 3B          DEC SP
0034 E1          POP HL
0035 E5          PUSH HL
0036 119800      LD DE,0098H
0039 19          ADD HL,DE
003A EB          EX DE,HL
003B 010C00      LD BC,000CH
003E CD3C20      CALL 203CH
0041 DD4E0B      LD C,(IX+0BH)
0044 DD460C      LD B,(IX+0CH)
0047 CD2B2D      CALL 2D2BH
004A CDE32D      CALL 2DE3H
004D 3A005B      LD A,(5B00H)
0050 FE03        CP 03H
0052 2821        JR Z,0075H
0054 FE00        CP 00H
0056 2839        JR Z,0091H
0058 E1          POP HL
0059 E5          PUSH HL
005A 11A400      LD DE,00A4H
005D 19          ADD HL,DE
005E EB          EX DE,HL
005F 010D00      LD BC,000DH
0062 CD3C20      CALL 203CH
0065 3A0E5B      LD A,(5B0EH)
0068 CBBF        RES 7,A
006A CBB7        RES 6,A
006C CBAF        RES 5,A
006E 2660        LD H,60H
0070 84          ADD A,H
0071 D7          RST 10H
0072 C3B01B      JP 1BB0H
0075 E1          POP HL
0076 E5          PUSH HL
0077 11B100      LD DE,00B1H
007A 19          ADD HL,DE
007B EB          EX DE,HL
007C 010D00      LD BC,000DH
007F CD3C20      CALL 203CH
0082 DD4E0D      LD C,(IX+0DH)
0085 DD460E      LD B,(IX+0EH)
0088 CD2B2D      CALL 2D2BH
008B CDE32D      CALL 2DE3H
008E C3B01B      JP 1BB0H
0091 DD460E      LD B,(IX+0EH)
0094 78          LD A,B
0095 E6C0        AND C0H
0097 2016        JR NZ,00AFH
0099 DD4E0D      LD C,(IX+0DH)
009C CD2B2D      CALL 2D2BH
009F E1          POP HL
00A0 E5          PUSH HL
00A1 11BE00      LD DE,00BEH
00A4 19          ADD HL,DE
00A5 EB          EX DE,HL
00A6 011500      LD BC,0015H
00A9 CD3C20      CALL 203CH
00AC CDE32D      CALL 2DE3H
00AF E1          POP HL
00B0 11D300      LD DE,00D3H
00B3 19          ADD HL,DE
00B4 EB          EX DE,HL
00B5 011900      LD BC,0019H
00B8 CD3C20      CALL 203CH
00BB DD4E0F      LD C,(IX+0FH)
00BE DD4610      LD B,(IX+10H)
00C1 CD2B2D      CALL 2D2BH
00C4 CDE32D      CALL 2DE3H
00C7 C3B01B      JP 1BB0H
00CA 0D          DEC C
00CB 0D          DEC C
```

```
5 GOTO 35
10 CLEAR 29999
20 FOR A=10 TO 20:BEEP 0.05,A:NEXT A
25 RESTORE
30 FOR B=30000 TO 30285:READ C:POKE B,C:NEXT B
35 PAPER 1:BORDER 1:INK 7:CLS
40 PRINT AT 11,6;"NYOMJ MEG EGY GOMBOT!"
45 PAUSE 0:CLS
50 RANDOMIZE USR 30000
60 SAVE "FEJLEC" LINE 10
70 DATA 243,17,17,0,175,55,221,33,0,91,205,86,5,48,242
80 DATA 251,221,33,0,91,205,107,13,62,2,205,1,22,58,0
90 DATA 91,17,192,9,205,10,12,6,10,17,0,91,19,26,215,16
100 DATA 251,205,124,0,59,59,225,229,17,152,0,25,235,1,12
110 DATA 0,205,60,32,221,78,11,221,70,12,205,43,45,205
120 DATA 227,45,58,0,91,254,3,40,33,254,0,40,57,225,229
130 DATA 17,164,0,25,235,1,13,0,205,60,32,58,14,91,203,191
140 DATA 203,183,203,175,38,96,132,215,195,176,27,225
150 DATA 229,17,177,0,25,235,1,13,0,205,60,32,221,78,13
160 DATA 221,70,14,205,43,45,205,227,45,195,176,27,221
170 DATA 70,14,120,230,192,32,22,221,78,13,205,43,45,225
180 DATA 229,17,190,0,25,235,1,21,0,205,60,32,205,227,45
190 DATA 225,17,211,0,25,235,1,25,0,205,60,32,221,78,15
200 DATA 221,70,16,205,43,45,205,227,45,195,176,27,13,13
210 REM SZOVEG KODJAI
220 DATA 66,121,116,101,115,32,68,98,46,58,13,13
230 DATA 80,114,111,113,114,46,32,78,101,118,58,13,13
240 DATA 69,108,115,111,32,32,66,121,116,101,58,13,13
250 DATA 66,65,83,73,67,45,105,110,100,105,116,97,115,105,
    32,115,111,114
260 DATA 58,13,13,65,32,66,65,83,73,67,45,112,114,111,103
270 DATA 114,97,109,32,104,111,115,115,122,97,58
```

A 30-as sorban történik a gépi kódú rutin beírása, amit majd a RANDOMIZE USR 30 000 utasítással hívhatunk meg. A rutin 204 bájt (70–200-as sorok) hosszúságú. A 220–270 DATA sorok a monitorra kerülő szöveg kódjait tartalmazzák.

A lista begépelése után a GOTO 60 utasítással magnóra rögzíthetjük programunkat, és csak ezután futtassuk. Ezzel elkerülhetjük, hogy az esetleges hiba miatt munkánk kárba vesszen. Ha MICRODRIVE kazettán kívánjuk megőrizni a programot, akkor a 60-as sort cseréljük ki a következőre:

60 SAVE * "m" ; 1; "FEJLEC" LINE 10,

Ha az így rögzített programot visszatöltjük, automatikusan elindul.

Egy gomb megnyomása után töltsük a gépbe a másolandó programrészletet. A fejléc leolvasása után megjelennek a monitoron a program adatai.

Ha egy következő blokkot akarunk másolni, előtte RUN paranccsal indítsuk újból programunkat.

A másolandó anyagról szerzett információk segítségével gazdaságosabban tölthetjük meg MICRODRIVE kazettánkat, és megtudhatjuk, hogy kódos programunkat a memória mely címétől kezdve kell kimásolnunk.

Leggyorsabban úgy másolhatunk, ha például a következő sort parancsként adjuk ki:

LOAD""CODE:SAVE*"m";1;"CIM"CODE

mezőcím db

25 000, 200

Ez a sor előbb beolvassa magnóról a programot, majd MICRODRIVE kazettára viszi.

A fenti módszer minden olyan esetben alkalmazható, ha van 1–2 kbájt szabad BASIC tárterületünk.

MIHÁLYFI JÁNOS

ZX81

Inverz képernyő

A program a képernyőt a RAND USR 16514 parancsra inverzbe írja át. Ezt a gépi kódú rutint nagy haszonnal lehet alkalmazni játékprogramokban. Például hibás adat vagy hasonló külső hiba esetén egy rövid ciklusba írva, ijesztő villogtatást produkál.

A beíráshoz szükséges BASIC program:

```
10 LET X=16514
20 INPUT A$
30 IF A$="S" THEN STOP
40 SCROLL
50 PRINT X,A$
60 LET A=16
   (CODE A$(1)-28) + CODE A$(2)-28
70 POKE X,A
80 LET X=X+1
90 GOTO 20
```

A hexakódok beadása, illetve a futtatás után egyenes parancsként beadjuk a következőt: POKE 16510,0. Ez változtatja át a REM sor címkéjét 0-ra, így nem lehet EDIT-elni.

A program azt használja fel, hogy egy karakter és inverzének kódja között 128 a különbség ABS(128). Így a képernyőfájl minden bájtjához hozzáad 128-at. A túlcsordulás esetével nem kell foglalkozni, hiszen a gép ezt is úgy kezeli, mintha a kivonás abszolút értékét írtuk volna fel.

A program ilyen formájában a képernyőnek az üzenetek részére fenntartott, alsó két sorát is inverzbe váltja. Ha ezt nem akarjuk, akkor POKE 16518,16-ot kell beírnunk.

GYÜRE PÉTER

Decimális cím	Hexakód	Mnemonik	Megjegyzés
16514	2A	LD, HL (NN)	A 16396 címen levő rendszerváltozó
5	0C		tartalmazza a DISPLAY FILE kezdőcímét
6	40	16396	
7	1E	LD, E, N	Ez a külső ciklus (24 képernyősor)
8	18	24 decimal	
9	23	INC HL	
20	16	LD D, N	A belső ciklus (soronként 32 karakter)
1	20	32	
2	3E	LD, A,N	Ez a tulajdonképpeni program,
3	80	128	a HL regiszterben tárolt aktuális
4	86	ADD A, (HL)	képernyőcím tartalmához 128-at ad,
5	77	LD (HL), A	ez lesz az inverz karakter kódja
6	23	INC HL	Egy képernyőhellyel továbblép
7	15	DEC D	
8	C2	JP NZ 16522	Ha még nincs az egész sor inverzben,
9	8A		akkor eggyel továbblép
30	40	16522	
1	10	DEC E	Ha az egész sor inverzben van, akkor egy sorral lejjebb lép
2	C2	JP NZ 16519	16519-re megy azért, hogy a képernyősor
3	87	16519	33. karakterét (NEW LINE (code 128))
4	40		kihagyja, nehogy a program elszálljon
5	C9	RET	Visszatér BASIC-be

COMMODORE 64

A LIST utasítás módosítása

A C64-es géphez leggyakrabban a Seikosha GP 100 és MPS 801-es nyomtatókat alkalmazzák. Ezeken a nyomtatókon nem állítható be automatikus lapdobás. Programból történő nyomtatás esetén azonban ez egy kis kényelmetlenséggel ugyan, de mégis megvalósítható.

Problémát jelent ellenben a hosszabb programok listázása. Szinte biztos, hogy legfontosabb programsoraink fognak a papír perforációjára esni, kellő mérgelődést okozva. Az alábbi program használata megkímél bennünket ettől a bosszúságtól.

És ha már úgyis hozzányúlunk a LIST alapszóhoz, kiküszöbölhetjük annak egy másik hibáját is. Az eredeti LIST utasítás nem alkalmazható csak parancs üzemmódban. Ezután például

```
100 FOR I=0 TO 3
110 LIST
120 NEXT
```

alakú programokat is használhatunk.

A rutin a kazettapufferben helyezhető el, de az assembler lista 1. sorának módosításával más szabad helyen is működtethető.

A RUN/STOP RESTORE gombok megnyomásával hatástalaníthatjuk a rutint, de SYS 828-cal újra indítható.

```
100 REM ** LIST MODOSITO PRG **
101 REM (C) TARNOKI LASZLO
102 FORI=0TO162
103 READX:SZ=SZ+X
104 POKE828+I,X:NEXT
105 IFSZ<>21172THENPRINT
    "HIBA VAN A DATA SOROKBAN!":END
106 SYS828:NEW
107 DATA 162, 32,169,160,160,  0,132
108 DATA 251,133,252,177,251,145,251
109 DATA 200,208,249,230,252,202,208
110 DATA 244,169,124,141, 66,160,169
111 DATA   3,141, 67,160,169,168,141
112 DATA 216,166,169,  3,141,217,166
113 DATA 169, 32,141,215,166,169,234
114 DATA 141,187,166,141,188,166,169
115 DATA  96,141, 20,167,169, 54,133
116 DATA   1, 96,169,  0,141,219,  3
117 DATA 141,220,  3,165,122, 72,165
118 DATA 123, 72, 32,121,  0, 32,156
119 DATA 166,169,  0,141,219,  3,141
120 DATA 220,  3, 32,215,170,104,133
121 DATA 123,104,133,122, 32,248,168
122 DATA  76,121,  0,140,221,  3,174
123 DATA 217,  3,236,220,  3,240,  6
124 DATA 238,220,  3, 76,210,  3,174
125 DATA 218,  3,236,219,  3,240,  9
126 DATA  32,215,170,238,219,  3, 76
127 DATA 185,  3,169,  0,141,220,  3
128 DATA 141,219,  3,172,221,  3,200
129 DATA 177, 95, 96
130 DATA  62:REM
    ** EGY LAPRA NYOMTATHATO SOROK
131 DATA  10:REM ** A PERFORACIONAL
    KIHAGYANDO SOROK
132 DATA   0,  0,  0,  0
```

Az ismertetett program 62 sor nyomtatása után 10 sort emel, de ez a perforációk távolságának függvényében természetesen megváltoztatható.

TÁRNOKI LÁSZLÓ

ZX-SPECTRUM

Miniprogram

A programozás során szükség lehet a képernyőn levő grafika egy meghatározott részének megváltoztatására, címeinek ismeretére. Közismert, hogy Spectrumon – a képernyőt három részletben használva – a 16384-es címmel kezdődően az egyes bájtok tartalmazzák a képelemadatokat. Első látásra elég nehézkesnek tűnik, hogy egy konkrét karakterpozíció összetartozó bájtjainak címét ismerni kell. Ilyenkor jól használható a következő miniprogram.

Az érthetőség kedvéért először a megjelenítést végezzük el. Legyen például a feladat egy 3×4=12 karakternyi terület berajzolása, amelynek a bal felső karaktere a 8. sorban, a 2. oszlopban helyezkedik el (PRINT AT 8,2). A rajzoláshoz a dec.60-at (BIN 00111100) használjuk, hogy az egy sorban levő karakterek elkülönüljenek egymástól.

```
10 LET a=256:LET b=32
20 FOR i=0 TO 7
30 LET s=18434
40 LET s=s+i*a
50 FOR j=0 TO 3
60 LET s=s+b
70 FOR k=0 TO 2
80 POKE s+k,60
90 NEXT k:NEXT j:NEXT i
ahol
s=18434 a kezdőcím a PRINT AT 8,2-nél,
j – négy egymást követő sor
k – három egymás melletti karakterhely
```

A program vizuálisan is jól követhetően, a jelölt helyekre a BIN 00111100-nek megfelelő szélességű négyszögeket rajzol.

Ha töröljük a 20-as és a 40-es sorokat, valamint a ":NEXT i"-t, és a 80-as sort módosítjuk: 80 PRINT s+k, akkor a 12 karakterpozíció első bájtjainak címét kapjuk meg. Ha a 20-as és a 40-es sor változatlanul marad az eredeti programban, és csak a 80-as sort módosítjuk, akkor valamennyi cím megjelenik a beíráshoz szükséges sorrendben.

Más lehetőségek is vannak, például "PEEK s" vagy "READ" és "DATA" felhasználása az eredeti programban.

NAGY KÁROLY

COMMODORE 64

Számalakítás

Munkám során gyakran előfordul, hogy egy számot szerkesztve, pozícióazonosan kell kiíratnom a képernyőre vagy a printerre. A bemenő számok, eredmények általában különböző formákban állnak rendelkezésre. A kívánt alak kialakításában segít a következő szubrutin.

A bemeneti paraméterek a következők:

VI$ – a vizsgálandó szám sztring alakban
V1 – a kívánt jegyek száma a vessző előtt
V2 – a kívánt jegyek száma a vessző után (−1 esetén a vesszőt is elhagyja)

A szubrutin a számot a VI$ változóban adja vissza a kívánt alakban.

Működés közben a szubrutin megvizsgálja, hogy a bemenő adat szám-e. Hiba esetén a HI$ változóban "HIBA" található. Ez a számok átalakításán kívül alkalmassá teszi a szubrutint a képernyőről beolvasott értékek ellenőrzésére és átalakítására is. A szubrutin csak a szám utáni első szóközig veszi figyelembe a változó tartalmát.

A jelenlegi állapotban a szubrutin az üres helyeket szóközökkel tölti fel. Ha a 340, 360, 370, 410 és 420-as sorokat kijavítjuk, ez változtatható „0"-ra.

Ha a szám exponenciális formájú, akkor az eredményt negatív exponens esetén 0., pozitív exponens esetén *. formában alakítja ki a rutin.

RICHTER JÖRG

```
1 REM ************************
2 REM *
3 REM *   F O P R O G R A M
4 REM *
5 REM ************************
6 :
10 DIM PR$(12)
20 PR$(0)="0":PR$(1)="1":PR$(2)="2":PR$(3)="3":PR$(4)="4":PR$(5)="5"
22 PR$(6)="6":PR$(7)="7":PR$(8)="8":PR$(9)="9":PR$(10)=".":PR$(11)=" "
23 PR$(12)="-"
25 INPUT"VI$=";VI$
30 INPUT"V1 =";V1
35 INPUT"V2 =";V2
40 GOSUB 100:IF HI$="HIBA"THEN PRINT HI$:GOTO 25
50 JO$=VI$
55 PRINT "EREDMENY=";JO$
57 PRINT "  HOSSZA=";LEN(JO$)
60 GOTO20
70 STOP
80 END
100 REM ******************************
110 REM *
120 REM * SUB: SZAMKIALAKITO
130 REM *
140 REM ******************************
150 HI$="":P=0:I3=12:IFV2=-1THENV2=0:P=1
160 IFV1>10ORV2>10THENHI$="HIBA":RETURN
170 L=LEN(VI$):IFVI$=""THENVI$="0":L=1
180 Q=0
190 FORI1=1TOL
200 MU$=MID$(VI$,I1,1)
210 IFMU$="E"THEN300
220 IFMU$<>" "THENQ=Q+1
230 IFMU$=" "AND Q>0THENVI$=LEFT$(VI$,I1-1):L=LEN(VI$):GOTO310
240 IFMU$=" "THEN 300
250 FORI2=0TOI3
260 IFMU$=PR$(I2)THEN290
270 NEXTI2
280 HI$="HIBA":RETURN
290 IFI2=10THENI3=9
300 NEXT I1
310 FORI1=1TOL
320 IF MID$(VI$,I1,1)="."THEN360
330 NEXTI1
340 VY$=RIGHT$("          "+VI$,V1)+"."+LEFT$("          ",V2)
350 GOTO430
360 VY$=RIGHT$("          "+LEFT$(VI$,I1-1),V1)+"."
370 VY$=VY$+LEFT$(RIGHT$(VI$,L-I1)+"          ",V2):GOTO430
380 I1=I1+1
390 IFMID$(VI$,I1,1)="-"THEN410
400 IFMID$(VI$,I1,1)="+"THEN420
405 HI$="HIBA":RETURN
410 VY$=RIGHT$("          ",V1-1)+"0."+RIGHT$("          ",V2):GOTO430
420 VY$=RIGHT$("          ",V1-1)+"*."+RIGHT$("          ",V2)
430 VI$=VY$
440 IFP=1THENVI$=LEFT$(VI$,LEN(VI$)-1)
450 RETURN
```

COMMODORE 64

Magyar nyelvű hibaüzenet

Ez a kis program azok számára lehet hasznos, akik nem tudnak angolul, és még nincs nagy gyakorlatuk a C64 kezelésében, de szeretnének komolyabb BASIC programokat írni. Begépelve vagy betöltve és lefuttatva a programot, nem kell többé minden hibaüzenet megjelenésekor a felhasználói kézikönyvet elővenni.

```
10 REM           MAGYAR HIBAUZENET
20 A=49152
30 PRINT"▮▮▮▮▮▮▮▮▮▮EGY KIS TURELMET"
40 REM
50 REM*******GEPI KOD BETOLTESE*********
60 REM
70 READB$:IFB$="@"THEN190
80 L$=LEFT$(B$,1):R$=RIGHT$(B$,1)
90 L=VAL(L$)
100 IFL=0ANDL$<>"0"THENL=ASC(L$)-55
110 R=VAL(R$)
120 IFR=0ANDR$<>"0"THENR=ASC(R$)-55
130 D=16*L+R
140 POKEA,D:A=A+1
150 GOTO70
160 REM
170 REM******UZENETEK BETOLTESE*********
180 REM
190 B=A+64
200 B=A+64
210 FORI=1TO32
220 H=INT(B/256):L=B-H*256
230 POKEA,L:POKEA+1,H:A=A+2
240 READB$
250 FORJ=1TOLEN(B$)
260 POKEB,ASC(MID$(B$,J,1)):B=B+1
270 NEXTJ
280 POKEB,0:B=B+1
290 NEXTI
300 POKE768,0:POKE769,192:GOTO790
310 REM
320 REM*********MAGA A PROGRAM***********
330 REM
340 DATA 8A,10,03,4C,74,A4,0A,AA,BD
350 DATA 55,C0,85,22,BD,56,C0,85,23
360 DATA 20,CC,FF,A9,00,85,13,20,D7
370 DATA AA,20,45,AB,A5,22,A4,23,20
380 DATA 1E,AB,20,7A,A6,A4,3A,C8,F0
390 DATA 26,AD,93,C0,AC,94,C0,20,1E
400 DATA AB,A5,3A,A6,39,85,62,86,63
410 DATA A2,90,38,20,49,BC,20,DF,BD
420 DATA 20,1E,AB,AD,95,C0,AC,96,C0
430 DATA 20,1E,AB,4C,74,A4,@
440 REM
450 REM**********UZENETEK**************
460 REM
470 DATA"TUL SOK FILE "
480 DATA"NYITOTT FILE "
490 DATA"A FILE NINCS MEGNYITVA "
500 DATA"NEM LETEZO FILE "
510 DATA"EGYSEG NINCS BEKAPCSOLVA "
520 DATA"NEM INPUT FILE "
530 DATA"NEM OUTPUT FILE "
540 DATA"HIANYZIK A FILENEV "
550 DATA"ERVENYTELEN EGYSEGSZAM "
560 DATA"NEXT FOR NELKUL "
570 DATA"SZINTAXIS HIBA "
580 DATA"RETURN GOSUB NELKUL "
590 DATA"NINCS TOBB ADAT "
600 DATA"NEM MEGENGEDETT MENNYISEG "
610 DATA"TULCSORDULAS "
620 DATA"MEMORIA MEGTELT "
630 DATA"UGRASI HIBA "
640 DATA"HIBAS TOMBINDEX "
650 DATA"TOMB UJRADIMENZIONALAS "
660 DATA"OSZTAS NULLAVAL "
670 DATA"PARANCS MODBAN NEM HASZNALHATO"
680 DATA"TIPUS HIBA "
690 DATA"TUL HOSSZU STRING "
700 DATA"FILE ADAT HIBA "
710 DATA"TUL BONYOLULT KIFEJEZES "
720 DATA"NEM FOLYTATHATO "
730 DATA"DEFINIALATLAN FUGGVENY "
740 DATA"VERIFY HIBA "
750 DATA"LOAD HIBA "
760 DATA"MEGSZAKITAS  "
770 DATA"A "
780 DATA". SORBAN"
790 DIMA(20):DIMA(30)
```

A program először a DATA sorokban tárolt gépi kódú programot tárolja le a C000 címtől kezdődően, majd folytatólag a szintén DATA sorokban tárolt hibaüzeneteket. A hibaüzenetek hossza lényegtelen, tetszőlegesen kicserélhetők, de a sorrendet nem szabad megváltoztatni.

Az előzőekből következik, hogy miután a program lefutott, kitörölhető, de a hatása megmarad mindaddig, amíg csak BASIC-ben dolgozunk, vagy nem írunk a C000-val kezdődő területre.

A program nem érinti az INPUT utasítás üzeneteit (például REDO FROM START), a bemeneti-kimeneti műveletek jelzéseit (SEARCHING FOR ..., LOADING stb.) és a „READY" feliratot.

SIMON LÁSZLÓ

COMMODORE 64

A képernyő-tartalom kinyomtatása

```
10 REM ***********************************
20 REM **   ,     .  , , ,              **
30 REM **  KÉPERNYŐ-KIÍRÓ PROGRAMM      **
40 REM **     COMMODORE C 64 - RE       **
50 REM ** ,               ,    ,    ,   **
60 REM ** IRTA:GLOETZER TAMÁS LÁSZLÓ **
70 REM **                               **
80 REM **  VIII. OSZT. TANULÓ; VÁC     **
90 REM ***********************************
100 D=1064
110 OPEN4,4: CMD4
120 FOR A= 1024 TO 2023
130 IF A=D THEN PRINT#4," ": D=D+40
140 B=PEEK(A)
150 E=B
160 IF B>128 THEN B=B-128
170 IF B>=0 AND B<32 OR B>96
    THEN C=B+64: GOTO 200
180 IF B>31 AND B<64 THEN C=B: GOTO 200
190 IF B>63 AND B<96 THEN C=B+32
200 C$=CHR$(C)
210 IF E>128 THEN PRINT#4,CHR$(18)
    ;C$;CHR$(146);: GOTO 230
220 PRINT#4,C$;
230 NEXT A
240 CLOSE 4
250 END
```

Gyakran előfordul, hogy képernyőnket változtatás nélkül szeretnénk kinyomtatni. Erre az esetre készítettem az alábbi programot, melynek rövid leírása a következő:

- 100 A D változó értéke a képernyő 2. sorának 1. karaktere
- 110 Nyitjuk a nyomtatót
- 120 Ciklus indul a teljes képernyőtartományra
- 130 Ha a ciklus sor elejére ér, akkor nyomtasson egy szóközt, és a sorszámláló a következő sor elejére mutasson
- 140 Olvassa ki a video-RAM megfelelő értékét
- 150 Bevezetjük az E segédváltozót
- 160–190 Rendezzük a kiolvasott értékeket és 200-ban CHR$-é alakítjuk
- 210 Ha a kiolvasott érték nagyobb 128-nál, akkor inverz karaktereket nyomtat ki
- 220 Egyébként normál üzemmódban nyomtat
- 240 Lezárjuk a nyomtatót
- 250 Ha szubrutinban használjuk ezt a programot, akkor ide természetesen RETURN kerül

Meg kell említeni, hogy a nyomtatás 40 karakteres lesz – amilyen a képernyő is –, tehát csak a papír bal fele lesz kihasználva.

GLOETZER TAMÁS LÁSZLÓ

ADOK – VESZEK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,— Ft, magánszemélyeknek az első sor 50,— Ft, minden további sor 20,— Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

• **KERESEM az ABERSOFT LTD.** 1983-ban megjelent Microdrive kompatibilis FORTH implementációját, valamint Spectrum-tulajdonosok ismeretségét programcsere céljából. Közel 300 programmal rendelkezem. Sári Gábor, Szolnok, Vörös Csillag út 76. XI. 1. 5000

• **COMMODORE 64** eladó datasettel, garanciával, vámkezelve. Ráczné, Csongrád, Dózsa tér 5.

• **ELADNÁM 1 KBÁJTOS ZX 81** mikroszámítógépemet, 16 kbájtos memóriabővítővel. Járóka László, Budapest, Szív u. 3–5. 1063

• **COMMODORE VC20** számítógép eladó. Ára 12 000,— Ft. Benkő Gábor, Budapest, Dózsa György út 108. II. em. 2. Telefon: 297-588

• **ZX 81** 16 kbájtos bővítménnyel eladó. Mezei György. Telefon: 184-703, 8–16.30-ig

• **VÁLLALOK:** személyiszámítógép-javítást és karbantartást! Közületek részére is! Pásztor Ferenc, Szolnok, Mátyás király u. 2. V. em. 3.

• **COMMODORE 64** típusú személyi számítógéphez mezőgazdasági üzemi igényeket kielégítő programokat vásárolunk. Fejér megyei TESZÖV Gazdaságpolitikai titkársága, 8000 Székesfehérvár, József Attila u. 55. III. em. Telefon: 22-14-055

• **ATARI 800XL-ES** 64 kbájtos személyi számítógép magnóval, játékkazettákkal és angol nyelvű leírásokkal eladó. Áhel Elemér, Felsőpáhok, Szabadság u. 41/b. 8395

• **ELADNÁM 1 DB VC–20 ÉS 1 DB ZX 81** számítógépemet, valamint az utóbbihoz 1 db 16 kbájtos tárolóegységet. Járóka László, Bp. Szív u. 3–5. fsz. 5. 1063

• **KÖZVETÍTJÜK** személyi számítógépek adásvételét. Commodore VC-20 és C16 mikrogépekhez magyar használati tájékoztatót, valamint hardver- és szoftvertermékeket kínálunk alacsony áron. ZX 81 számítógépek már 6000 Ft-ért vásárolhatók! Kérje tájékoztatónkat! INTER GM. Kaposvár, Rákóczi tér 1. 7400

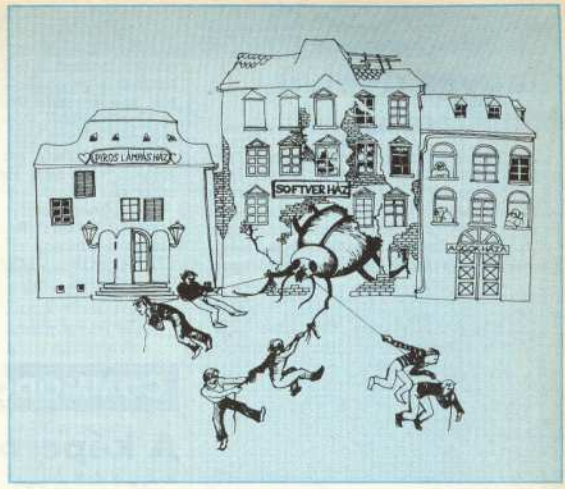
• **VEGYE IGÉNYBE SZOLGÁLTATÁSAINKAT!** Olcsó óradíjért használhatja 48 k-s Spectrum gépeinket. Szoftvercsere! Folyamatosan indulnak BASIC és Assembler tanfolyamok. Eredményes angol nyelvoktatás számítógéppel! Spectrum GM. Bp. I., Hattyú u. 4. Telefon: 161-305. Nyitva: 13–19 óráig, szombat-vasárnap is.

T. SZERZŐ KOLLÉGÁK!

A honoráriumok zökkenőmentes átutalása érdekében kérjük, hogy az írásokkal, cikkekkel együtt az alábbi adatokat is szíveskedjenek szerkesztőségünkbe elküldeni:

Név
Születési hely, idő
Anyja neve
Lakáscím, telefonszám
Munkahely megnevezése
Munkahelyi címe, telefonszáma
Személyi száma
Utalás esetén melyik címre kéri a honoráriumot

Tanulságos kalandok a számítástechnika alkalmazásai során



Az olcsó mikrogépek megjelenése óta a „személyi számítástechnika" ragályszerű sebességgel terjed. Az „élenjáró tudomány és technika" e téren ma valóban hozzáférhetővé vált a *„széles tömegek"* számára, a tudásszomjas iskolástól kezdve, a számítástechnika hivatásos műveléséből túlzott alkalmatlansága vagy túlzott alkalmassága miatt kiszorítottakon keresztül, a lelkiismeretes és lelkiismeretlen művelőkig mindenkinek. Az eszközök olcsósága és elérhetősége révén a „személyi számítástechnika", ha a népi kohászathoz hasonló kampányt talán nem is fog létrehozni, *népi számítástechnikát* azonban minden bizonnyal.

A jelekből ítélve a „népszámológép" kultusza és a „népi számítástechnika" nem minden téren fogja elérni például a népművészet és a népi gyógyászat megbecsültségét. (Ez azonban nem a „hivatalos" színvonal túlságosan magas volta miatt lesz így.)

A számítástechnikai anekdotákkal azonban más a helyzet. Jellemző, hogy amig egy-egy valóban értékes eredményecske felbukkanására éveket kell várnunk, a hol humoros, hol elszomorító, de mindig tanulságos számítástechnikai történetekből rövid idő alatt olyan bő termés halmozódott fel hazánkban, hogy az a nagy számítástechnikai hagyományokkal rendelkező országoknak is becsületére válna.

Úgy látszik, hogy a számítástechnikát csinálni még nem, de róla intelligensen csevegni, azt bírálni, „fogjuk meg és vigyétek" módon utat mutatni már sokan megtanultak.

Úgy gondolom, hogy nem érdektelen, ha idejében felhívjuk a figyelmet a számítástechnikai anekdotakincs gyűjtésének fontosságára. Ezt nemcsak tudományszociológiai, nemcsak szórakoztatási szempontok, hanem a szakmai érdekek is indokolják, hiszen e történetek – érdekes módon – kivétel nélkül szakmai tanulságokat is tartalmaznak.

Annak szemléltetésére, hogy milyen történetekre gondolok, leírok néhányat saját tapasztalatomból. Mielőtt azonban ezekre rátérnék, érdemes felhívni a figyelmet arra is, hogy az élesebb szeműek és türelemmel is rendelkezők hasznos és szórakoztató szakmai anekdotákon kívül mást is gyűjthetnek. Például sohasem helyesbített szakmai hibákat és nyilvánvaló plágiumokat a különböző műsorokból és a sajtóból. Néhány év múlva döbbenetes erejű kortörténet (és kórtörténet) lesz például a mai számítástechnikai hirdetéstömeg és a vásári és egyéb számítástechnikai prospektusok gyűjteménye is.

Éves periódus a nyersanyag-összetételben

Egy – bányászott nyersanyagot feldolgozó – ipari üzemben valamilyen őrlemény összetételének „állandó értéken" tartása volt a feladat. Az összetételre vonatkozó mérési adatok több évre visszamenően kifogástalan formában és tartalommal rendelkezésre álltak, de érdemi tanulmányozásnak még nem vetették alá azokat.

E sorok írója vette magának a fáradságot, és a mérési adatok idősorait grafikusan ábrázolta. Megérte. Szokatlan jelenség került felszínre. Az ábrákról ugyanis szinte kiabált az éves periodikusság. Éves periódusú változása – ez ilyenkor a legtermészetesebb ötlet – az időjárásnak van. A Meteorológiai Intézettől az üzem környezetének csapadék- és hőmérsékletadatait beszerezve, az a megdöbbentő tény is nyilvánvalóvá vált, hogy szoros lineáris korrelációt van a bányászott nyersanyag összetétele és a csapadék között.

Megkezdtem a munkát a jelenség okának kiderítésére, ám nem sok sikerrel. Feltételezéseim sorra kudarcot vallottak. A részletes elemzés során ki kellett zárni például azt a lehetőséget, hogy a csapadék úgy változtatja meg a bányászott nyersanyag összetételét, hogy kimos valamennyit valamelyik összetevőből.

A víz valamiféle beépülése sem magyarázta a jelenséget, mert az anyag összetételének laboratóriumi vizsgálatát előzőleg kiizzított, száraz mintán végezték.

Végül a kutatási jelentésben a problémát csupán felvetni, megfogalmazni tudtam, megoldani azonban nem.

A megrendelő képviselője ennek ellenére meg volt elégedve az eredménnyel, és közölte, hogy őket is meglepte a csapadék és a nyersanyag-összetétel között, általam észrevett, addig ismeretlen, érdekes összefüggés. Azután pedig mosolyogva mondta: „A jelenség okát azonban – fogadni merek – még csak nem is sejted".

Valóban. Ilyen okra rájönni – különösen távolból – nem is olyan könnyű, és mint azt közölték, ez a megrendelő szakembereinek sem ment egyik pillanatról a másikra.

Az ok a következő volt. Száraz időben a nyersanyag-kitermelés – a szállítási és egyéb munka minimalizálása miatt – egy olyan, a feldolgozás helyéhez közeli helyen folyt, amelyről a kibányászott nyersanyagot csak egy mélyen fekvő teknőn keresztül vezető úton lehetett elszállítani. Csapadékos időben e teknőt ellepte a víz, lehetetlenné téve a szállítást. Ilyenkor egy távolabbi helyen kellett folytatni a fejtést, ahonnan a szállítás hosszabb úton ugyan, de megoldható volt. E távolabbi helyen azonban a bányászott nyersanyag összetétele lényegesen eltért a közelebbi helyen bányászottétől. Így „vezérelte" tehát a csapadékmennyiség a bányászott nyersanyag összetételének alakulását.

Ezek után – talán felesleges hangsúlyozni – egészen más alapokon kellett a számológépes irányítórendszert megtervezni.

Számomra ez életre szólóan tanulságos, józan óvatosságra intő kaland volt. Szerencsére az irányítórendszerre vonatkozóan még nem készült túlságosan mély – és költségvonzatok szempontjából végzetes – javaslat.

Egy „reménytelenül" nehéz differenciálhányados-kiszámítási feladat

Egy hőhatásra vegyi és egyéb okokból térfogatát változtató, inhomogén szerkezetű rendszer (például ékszíj, gumiabroncs) gyártásában nehézségek léptek fel, az egyes alkotórészek egymástól eltérő térfogatváltozásából adódóan.

Az illetékes szakemberek gondosan elemezték a helyzetet és az elvégzendő feladatot, majd – mint ilyenkor szokásos – „optimalizáltak". Munkájuk egy egyváltozós függvény szélsőértékének kiszámítási feladatába torkollott. Ha ezt sikerült volna megoldaniuk, akkor minden el lett volna intézve. A függvény azonban rendkívül bonyolult volt, és olyan hosszú, hogy – amint mondani szokták – „lelógott az asztalról".

Ennek a függvénynek keresték a differenciálhányadosát, hogy utána majd annak gyökeit megkeresve, a szélsőértékeket is meghatározzák.

Igen ám, de ezen az úton nem jutottak megnyugtató eredményre, mert mindenki derivált ugyan, aki csak tudott deriválni a vállalatnál, de nem akadt a végeredmények között két egyforma. Így hát nem volt mit tenni – rákényszerültek –, felkeresték a matematikust is. Imigyen: „Tudsz-e deriválni?" A kérdés nagyon gyanús volt, ezért kitérésképpen visszakérdeztem: „Mi a feladat?" Válaszként persze nem a teljes feladatot, hanem a deriválandó függvényt mondták meg.

Ránézésből meg lehetett állapítani, hogy a differenciálási feladat ugyan rendkívül primitív, de a függvény szokatlan hosszúsága és bonyolultsága miatt reménytelennek látszik számolási hiba nélkül megúszni a dolgot. Másnapra* ígértem a választ. Mondanom sem kell, hogy a sok órás kellemetlen számolásból csak azt a tapasztalatot szűrhettem le, hogy ahány kísérlet, annyi különböző végeredmény.

Másnap tovább puhatolóztam a feladat (a teljes feladat!) után. Ekkor közölték, hogy a deriváltnak még a gyökeire is szükség van. Közben ők is dolgoztak, és közmegegyezéssel megállapodtak (!) egy differenciálhányadosban, de a gyökök megkeresésébe belefutott a bicskájuk.

A zéruspontoknál gyanút fogtam. „Az meg minek?" – kérdeztem. „Hát a szélsőértékhez" – volt a válasz.

Most már teljesen tiszta volt előttem a helyzet. Kijelentettem: „Azt nem úgy kell megoldani. Mondjátok el az egész problémát!" Elmondták. Kiderült például, hogy a sok szinusz meg koszinusz szorzat egyszerű vetítésekből és területszámításokból jött be a függvénybe.

Megkérdeztem, hogy milyen pontossággal szükséges az eredmény, és azt is, hogy az egyes állandók és segédváltozók milyen intervallumban mozoghatnak. A pontossági igény nem volt különösen nagy. A közölt változási intervallumok ismeretében a függvény valódi értelmezési tartománya jelentősen összeszűkült. (A gyakorlatban egy függvény értelmezési tartománya ugyanis csak nagyon ritkán azonos a matematikai legbővebb értelmezhetőségi tartománnyal. A legtöbb esetben – meggondolatlanul – a szükségesnél bővebb értelmezési tartománnyal dolgozunk, nemegyszer tetemes mennyiségű felesleges munkát végezve és végeztetve.)

A kívánt pontosság és a szűkebb értelmezési tartomány ismeretében döntöttem. Odaálltam egy rajztábla elé, amelyen A0 formátumú, sima papírra,

jó hegyes, 7H-s ceruzával – ügyelve a pontosságra – megrajzoltam a tervezett alakzat egy lehetséges (de persze legfeljebb csak véletlenül „optimális") megoldását. Ezen a rajzon gondos méréseket végeztem. Kiderült, hogy rajzom tényleg nem „optimális". A mérésekből és rövid és könnyű számolásból meghatározható volt a szükséges változtatások iránya. Ugyanerre a papírra elkészítettem egy másik változatot, amely szándékosan egy kicsit „túllőtt a célon". Újra mértem és számoltam. Majd pedig interpolációk után újra rajzkészítés következett. Az így kapott változat már majdnem jó volt, csak egyszer kellett módosítanom. Végül is sikerült az „optimumot" mintegy 50-szeres nagyításban – megkövetelt pontossággal – megrajzolni. Az elért pontosság a megkövetelt 0,1 mm-es hibakorláton egyértelműen belül volt.

Közöltem az eredményt. Sajnos – akkor még tapasztalatlan lévén – elárultam a módszert is.

A reakció nagyon lehangolt. „Rajzolni mi is tudunk." – volt a hűvös válasz, és éreztetett megvetéssel faképpel hagytak. (Az eredményt azért magukkal vitték.)

Ha deriváltam volna, egészen biztosan felnéztek volna rám. Beszéltek volna a rekordról, hogy milyen hosszú függvényt sikerült deriválnom. (A deriválási világrekordot nyilván kilométerben fogják mérni!) Így azonban, mivel csak a problémát oldottam meg – mégpedig nem a begyökeresedett előítéletek szerinti módon – figyelemre sem méltattak.

Ekkor tudatosodott bennem először, hogy a megrendelő megelégedése és a feladat legjobb megoldása két nagyon is különböző dolog.

Egy fogorvos ismerősöm mesélte. „Ha minden igyekezetemmel elértem, hogy ne törjön el a kihúzandó korhadt fog, és egy pillanat alatt kiemelem, úgy, hogy a beteg észre sem veszi, az az ő szemében nem teljesítmény. De ha ügyetlen vagyok és eltörik a fog, és több darabban kell kivésni, utána pedig varrni a foghúst (minél több öltés, annál jobb), akkor az teljesítmény, azt el tudja a beteg mesélni, tud vele dicsekedni, akkor meg vannak velem elégedve."

Szállítási „optimum"

Egy vállalat szállítási részlegétől, amely nagyszámú tehergépkocsival fuvaroz, természetes módon elvárják a vezetők, hogy az áru elszállítását minél kevesebb költséggel oldja meg.

A szállítási igazgatóság kerített is egy vállalkozót, aki – jó pénzért – „optimalizált". Munkája eredménye hozzám is eljutott. Nekem kellett volna megírni a programokat és „gépre tenni" a „rendszert". (Ezt nem vállaltam, ki fog derülni, hogy miért.)

Az egészben az volt a legmegdöbbentőbb, hogy a javaslatban, amit mindenki elfogadott és ami szerint az „optimalizálás" elvégeztetett, a következő – szállítási vállalatoknál széltében-hosszában használt – mutató kapta a főszerepet, mégpedig egyedül és kizárólagosan. A mutató a következő hányados volt:

$$\frac{\text{rakott km-ek}}{\text{rakott km-ek} + \text{üres km-ek}}$$

A felkért ámítástechnikus kreált egy algoritmust, amivel a szállítási terveket – úgymond – „javítani" lehetett, azaz egyre jobbakat lehetett szerkeszteni. Ezeknek az egyre „jobb" szállítási terveknek az volt a jellemzőjük, hogy a mutató értéke esetükben egyre nagyobb volt.

Az eljárás csak arra vigyázott, hogy a gépkocsik útvonala a kiinduló telephelyen kezdődjön és végződjön, és az útvonalak állomásai az összes feladó és célállomás halmazából kerüljenek ki. A javítást – ez a balhit már akkor széles körű elterjedtségnek örvendett – akkor kellett abbahagyni, ha akadt egy olyan javítás, amely egy előre rögzített számnál kisebb mértékben növelte a mutató értékét. „Nincs is ennél jobb megoldás" – állapította meg mindenki, hiszen az »üres furikázásra« használt benzin tiszta pénzkidobás, és ha több az üresen megtett kilométer, az rontja a mutató értékét, míg ha kevesebb, az javítja".

Szerencsére a napi szállítási tervek mégsem a szóban forgó algoritmus szerint készültek, de nem azért, mert kiderült a hiba, hanem mert az ámítástechnikus újabb és újabb követelésekkel állt elő a gépi programok elkészítéséért és a rendszer üzembe helyezéséért, amit a megrendelő már nem győzött anyagilag. Így került az egész probléma hozzám.

A kérdés, hogy hol követett el hibát a számítástechnikus, amivel kiérdemelte, hogy megfosszuk az sz betűtől, amit úgyis csak eredeti hivatásának elleplezésére használt. A választ – nem túl nehéz fejtőrőként – az olvasóra hagyjuk.

A kaland vége is érdekes. Nekem gépre kellett volna vinnem a „rendszert". Amikor a modellel kapcsolatos súlyos aggályomnak hangot adtam, a válasz egyértelmű és ellentmondást nem tűrő volt. „Az egész világon lineáris programozást alkalmaznak. Mi sem akarunk okosabbak lenni." Ez utóbbi törekvésüket nyilván nem volt nehéz megvalósítaniuk. A lineáris programozásnak pedig csak annyi szerepe volt, hogy egy kezdeti közelítő megoldás előállítására szolgált, ami természetesen csak véletlenül lehetett „optimális". Ezt a kezdeti közelítést „javították" (valójában rontották el véglegesen) az eljárással.

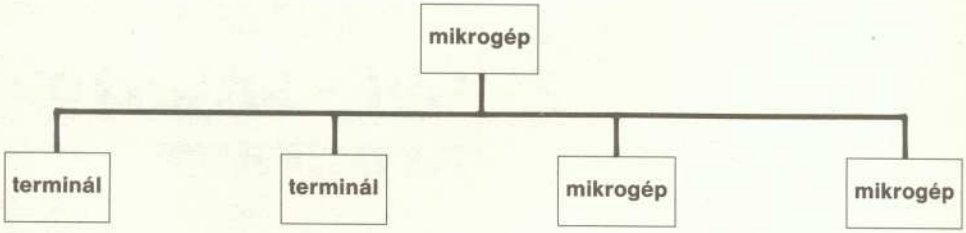
E sorok írója akkor a rugalmas elszakadás taktikáját választotta. És ilyen esetekben máig sem tud jobb tanácsot adni: menekülni, amíg nem késő.

POGÁNY CSABA

•

Hasonló, tanulságos és eredeti anekdotákat szívesen közzéteszünk a magazinban, ha azok nem személyeskedők és nem egyes személyek kultuszát szolgálják. (A szerk.)

ÚJ OPERÁCIÓS RENDSZEREINKET ajánljuk az I8080/85 és Z80-as mikrogépek felhasználóinak!



A MIREAL és MINET operációs rendszerek lehetővé teszik a központi gépre több terminál és több önálló mikrogép rákapcsolását. A mikrogépek között olyan logikai kapcsolat van, hogy egymás erőforrásait használhatják. A munkahelyek mindegyike CP/M vagy MP/M kompatibilis.

Megtekinthető a SW.'86 kiállításon a MIKROPO Gmk standján.

MIKROPO SZÁMÍTÁSTECHNIKAI GMK.

Budapest, VI., Nagymező u. 51.
1325 Bp. Pf. 52. Tel.: 325-768

Építsünk számítógépet! X.

Az előző részben a mikroprocesszor és a külvilág adatvonalait ismertettem.

A megfelelő címvonalak közül a felső nyolc (A15–A8) a processzor P47–P40 vonalain keresztül kapcsolódik az IC 4,5 (74367) bemeneti pontjaira, majd onnan a buszra. Az alsó nyolc egy multiplexer egyik oldalára (IC 7, 74LS373) adja a processzor P37–P30 vonalait, majd a buszra. A multiplexer, ahogy erről már szóltam, az adatok továbbítására is szolgál. Annak kiválasztása, hogy adat vagy cím kerüljön-e a multiplexerre, az AS (address select – címkiválasztó) vezetékkel történik, amely a mikroprocesszor SCl vonala. A címzőrendszer az *1. ábrán* látható.

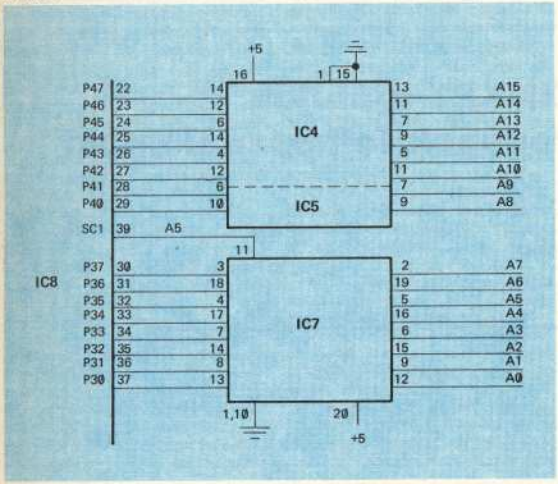
A vezérlőjelek, amelyeket a processzor a buszra kiad, a következők: a két órajel (Ø1, Ø2), amelyekről az előző cikkben már írtam; kiegészítésül annyit, hogy az E-jel, amely az órajelek alapja, a 4.9152 MHz frekvenciájú jeleket szolgáltató kristály frekvenciájának negyedelésével áll elő. A buszra adott vezérlőjelek közé tartozik az írás/olvasás vezérlő R/W̄ jel is, amelyről már szintén szó volt az előző részben. Az órajel-előállító részt a *2. ábra* mutatja. Végül egy eddig még nem említett jel a soros kommunikáció alapfrekvenciáját szolgáltató jel, a processzor P24 vonalán megjelenő szoftverrel beállítható frekvenciájú jel, amelyet az IC 2-vel kapuzva viszünk a buszra. A beállítható frekvenciák értéke az E-jel 16, 128, 1Ø24, 4Ø96-tal történő osztásából adódik, azaz 76ØØ, 96ØØ, 12ØØ, 3ØØ Baud lehet. Ez a rész a *3. ábrán* látható.

A mikroprocesszor működését a két megszakításjel (NMI, IRQ, *4. ábra*), a két alaphelyzetbe vivő jel (MRESET, RESET, *5. ábra*), a mikroprocesszor üzemmódját beállító jelek (P2Ø, P21, P22 vonalakra adott jelek, *6. ábra*) vezérlik a busz felől. A PORT 2-nek erről a funkciójáról a mikroprocesszort ismertető részben már volt szó.

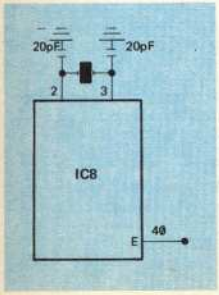
A kártya további funkcióiról a következő részben írok.

DR. SIMONYI ENDRE

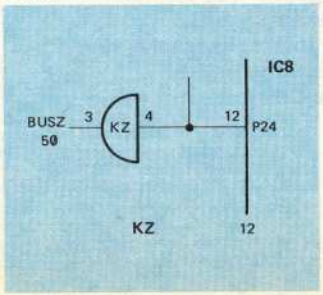
1. ábra

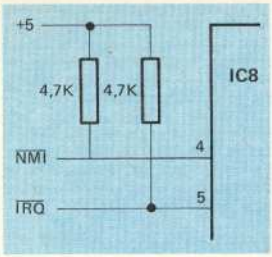


2. ábra

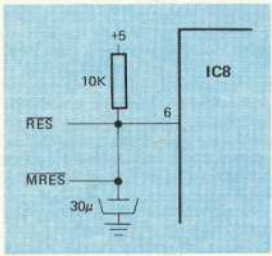


3. ábra

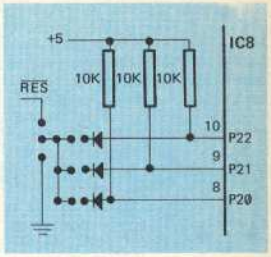




4. ábra



5. ábra



6. ábra

Kazettás nyilvántartó program

A C64-re irt program elsősorban programjaink nyilvántartására készült, de bármilyen sorszámból, szöveges adatból és numerikus adatból álló rekordsorozat nyilvántartására alkalmas.

Használatához csak annyit: betöltjük és indítjuk. Minden egyéb információt a program megad. Ezt elsőként az „étlap"-pal teszi, majd a választás után kérdez. Új programnak a listára bevitelénél például kiírja az adat sorszámát (8. ADAT BEVITELE), kérdezi a program nevét, azonosítószámát, majd visszatér az „étlap"-hoz.

Azok számára, akik érteni vagy módosítani kívánják a programomat, elláttam azt a szükséges számú megjegyzésekkel.

GÁBOR ANDRÁS

```
PRESS PLAY ON TAPE

 10 PRINT"■";POKE53280,0:POKE53281,0:GOSUB 10000
 20 PRINT"□        PROGRAM MENU 2 KRZETTARA"
 30 PRINT"□        ------------------------"
 40 PRINT"          IRTA ORBOR AMRAS 1985"
 60 PRINT"■■        ■■EVITEL"
 70 PRINT"■         ■TMORLES"
 80 PRINT"■         ■LMISTA NYOMTATORA"
 90 PRINT"■         ■FMELVETEL KRZETTARA"
100 PRINT"■         ■DMETOLTES KRZETTAROL"
105 PRINT"■         ■VMISSZA KERESES"
110 PRINT"■         ■LMISTA KEPERNYORE"
115 PRINT"■         ■HMADAT JAVITAS"
117 PRINT"■         ■SMORBA RENDEZES ABC SZERINT"
120 GET ME$:IF ME$="" THEN 120
130 IF ME$="B" THEN GOTO 1000
140 IF ME$="T" THEN GOTO 1500
150 IF ME$="L" THEN GOTO 2000
160 IF ME$="F" THEN GOTO 2500
165 IF ME$="E" THEN GOTO 2750
170 IF ME$="V" THEN GOTO 3000
180 IF ME$="I" THEN GOTO 3500
190 IF ME$="A" THEN GOTO 4000
200 IF ME$="S" THEN GOTO 5000
210 GOTO 120
1000 REM *** ADAT BEVITEL ***
1010 PRINT"■";CO;"■L ADAT BEVITELE"
1020 INPUT"MI A PROGRAM NEVE";PRO$(CO)
1030 INPUT"MI AZ AZONOSITOSZAMA(ASZ)";PRG(CO)
1040 CO=CO+1
1050 GOTO 20
1500 REM *** ADAT TORLES ***
1510 PRINT"□  ■BRATOK TORLESE"
1520 INPUT"HANYADIK PROGRAM";CL
1530 IF CL>CO THEN PRINT"NEM LETEZO ADAT!":FOR IM=0 TO 400:NEXT:GOTO 20
1540 FOR CE=CL+1 TO CO
1550 PRO$(CE-1)=PRO$(CE)
1560 PRG(CE-1)=PRG(CE)
1570 NEXT CE
1580 CO=CO-1
1590 GOTO 20
2000 REM *** LISTA NYOMTATORA ***
2010 PRINT"□  ■LISTAZRS NYOMTATORA■"
2015 PRINT"■                  GABOR ANDRAS'TT"
2020 INPUT"■A TULAJDONOS NEVE";NAME$
2030 INPUT"MILYEN EV";VE
2040 INPUT"MILYEN HONAP";MO$
2050 INPUT"HANYADIKA";DA
2060 OPEN4,4:CMD4
2070 PRINTCHR$(145)
2080 PRINTCHR$(14);:PRINTNAME$:PRINTCHR$(13);
2090 PRINTCHR$(15);:PRINTCHR$(18);:PRINTVE;MO$;DA:PRINTCHR$(146);
2100 PC=INT(CO/2)
2110 PRINTCHR$(13)CHR$(13)
2115 A$=""
2120 FOR I=1 TO PC
2125 A$=""
2130 IF(C10THENA$=" "
2131 IF(C100THENA$=" "+A$
2132 IF(C1000THENA$=" "+A$
2140 A$=A$+STR$(I)+"    "
2145 A$=A$+PRO$(I):FORJ=LEN(A$)TO30:A$=A$+" ":NEXT
2150 A$=A$+STR$(PRG(I)):FORJ=LEN(A$)TO40:A$=A$+" ":NEXT
2160 IFI+PC<10THENA$=A$+" "
2161 IFI+PC<100THENA$=A$+" "
2162 IFI+PC<1000THENA$=A$+" "
2170 A$=A$+STR$(I+PC)+"    "
2175 A$=A$+PRO$(I+PC):FORJ=LEN(A$)TO70:A$=A$+" ":NEXT
2180 A$=A$+STR$(PRG(I+PC))
2181 PRINTA$
2182 NEXT
2183 A$=""
2185 IF CO/2=INT(CO/2) THEN 2210
2190 IFCO<10THENA$=" "
2191 IFCO<100THENA$=" "+A$
2192 IFCO<1000THENA$=" "+A$
2193 A$=A$+STR$(CO)+"    "
2194 A$=A$+PRO$(CO):FORJ=LEN(A$)TO30:A$=A$+" ":NEXT
2195 A$=A$+STR$(PRG(CO)):FORJ=LEN(A$)TO40:A$=A$+" ":NEXT:PRINTA$
2210 PRINT#4," ":CLOSE4
2220 GOTO20
2500 REM *** FELVETEL KRZETTARA***
2510 PRINT"□  ■FELVETEL KRZETTARA■"
2520 PRINT"■ A MAGNO KESZ"
2530 GET ET$:IF ET$<>"I" THEN 2530
2535 PRINT"NYOMJA MEG A 'RECORD' ES A 'PLAY'GOMBOT■"
2540 OPEN1,1,1,"DATAS"
2550 PRINT#1,CO
2560 FOR I=1 TO CO-1
2570 PRINT#1,PRG$(I):PRINT#1,PRG(I)
2580 NEXT
2590 CLOSE 1
2600 PRINT"■":GOTO 20
2750 REM *** BETOLTES KRZETTAROL ***
2760 PRINT"□  ■BETOLTES KRZETTAROL■"
2870 PRINT"■ A MAGNO KESZ"
2880 GET ET$:IF ET$<>"I" THEN 2880
2885 PRINT"NYOMJA MEG A 'PLAY' GOMBOT■"
2890 OPEN1,1,0,"DATAS"
2900 INPUT#1,CO
2910 FOR I=1 TO CO-1
2920 INPUT#1,PRO$(I):INPUT#1,PRG(I)
2930 NEXT
2940 CLOSE 1
2950 PRINT"■":GOTO 20
3000 REM *** VISSZKERESES ***
3010 PRINT"□  ■VISSZKERESES"
3020 PRINT"■■■EV SZERINT,■■ZAM SZERINT,■■SZ SZERINT"
3030 GET ET$:IF ET$<>"N" AND ET$<>"S" AND ET$<>"A" THEN 3030
3040 IF ET$="N" THEN GOTO 3200
3042 IF ET$="A" THEN GOTO 3400
3045 REM             ** SZAM SZERINT **
3050 INPUT "HANYADIK PROGRAM";RS
3060 PRINT "     PRG. NEVE          ASZ"
3070 PRINTRS;TAB(5)PRO$(RS);TAB(25)PRG(RS)
3080 PRINT"UJRT(I/N)?          "
3090 GET ET$:IF ET$<>"I" AND ET$<>"N" THEN GOTO 3090
3100 IF ET$="N" THEN GOTO 20
3110 INPUT "HANYADIK PROGRAM";RS
3120 PRINT"□I              ":PRINT"□            ":PRINT"□          T"
3130 GOTO 3070
3200 REM              ** NEV SZERINT **
3210 INPUT"MI A PROGRAM NEVE";RS$
3220 FOR RE=1 TO CO
3230 IF PRO$(RE)=RS$ THEN PRINT RE;". NAME:"PRO$(RE)"  ID:";PRG(RE):GOTO 3260
3240 NEXT
3250 PRINT"NEMLETEZO PROGRAM NEV !":FOR I=0 TO 600:NEXT:GOTO 20
3260 PRINT"UJRT(I/N)?          "
3270 GET ET$:IF ET$<>"I" AND ET$<>"N" THEN GOTO 3270
3280 IF ET$="N" THEN GOTO 20
3290 PRINT"□          ":PRINT"□          ":PRINT"□          T":GOTO3210
3400 REM              ** ASZ SZERINT **
3410 INPUT"MI A PROGRAM AZONOSITOSZAMA";RS
3420 FOR RE=1 TO CO
3430 IF PRG(RE)=RS THEN PRINT RE;". NEV:";PRO$(RE);" ASZ:";PRG(RE):GOTO 3460
3440 NEXT RE
3450 PRINT"NEMLETEZO ASZ !":FOR I=0 TO 400:NEXT:GOTO 20
3460 PRINT"UJRT(I/N)?          "
3470 GET ET$:IF ET$<>"I" AND ET$<>"N" THEN GOTO 3470
3480 IF ET$="N" THEN GOTO 20
3490 GOTO 3410
3500 REM *** LISTA KEPERNYORE ***
3510 PRINT"□  ■PROGRAM LISTA■"
3520 PRINT"SZAM      PRG. NEV          ASZ"
3530 SC=1
3540 FOR I=1 TO CO-1
3550 IF SC/19=INT(SC/19) THEN GOTO3560
3555 GOTO3590
3560 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■TOVABB?"
3565 GET ET$:IF ET$="" THEN 3565
3566 IF ET$="Y" THEN 20
3567 IF ET$="■" AND I<(CO-19) THEN I=I+19
3568 IF ET$="■" AND I>19 THEN I=I-38
3570 IF ET$="■" AND I>38 THEN I=I-57
3572 IF ET$="■" AND I<(CO-38) THEN I=I+38
3575 PRINT"□  ■PROGRAM LISTA■"
3580 PRINT"SZAM      PRG. NEV          ASZ"
3590 PRINT"                    .....T"
3592 PRINT" ";I;TAB(7)PRO$(I);TAB(28);PRG(I)
3595 SC=SC+1
3600 NEXT
3610 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■VISSZA?"
3620 GET ET$:IF ET$="" THEN 3620
3630 GOTO 20
4000 REM *** JAVITAS ***
4010 PRINT"□  ■ADAT JAVITAS■"
4020 PRINT"■■SZ VAGY ■■EV JAVITAS?"
4030 GET ET$:IF ET$<>"A" AND ET$<>"N" THEN 4030
4040 IF ET$="A" THEN 4300
4042 IF ET$="N" THEN 4050
4045 GOTO 4030
4050 PRINT"KERESES ■■EV SZERINT,■■SZ SZERINT,VAGY    ■■ZAM SZERINT?"
4060 GET ET$:IF ET$<>"N" AND ET$<>"S" AND ET$<>"A" THEN 4060
4070 IF ET$="S" THEN GOTO 4160
4080 IF ET$="A" THEN GOTO 4210
4090 REM *** NEV JAVITAS ***
4100 INPUT"A REGI NEV";RRG$
4110 INPUT"AZ UJ NEV";PPG$
4120 FOR I=1 TO CO
4130 IF PRG$(I)=RRG$ THEN PRO$(I)=PPG$:GOTO 20
4140 NEXT
4150 PRINT"NEMLETEZO PROGRAM NEV !":FOR L=0 TO 600:NEXT:GOTO 20
4160 REM *** SZAM JAVITAS ***
4170 INPUT"HANYADIK PROGRAM";EN
4175 IF ENOCO THEN PRINT"TUL NAGY SZAM !":FOR L=0 TO 600:NEXT:GOTO 20
4180 PRINT"A REGI NEV";PRG$(EN)
4190 INPUT"AZ UJ NEV";PPG$
4200 PRG$(EN)=PPG$:GOTO 20
4210 REM *** ASZ JAVITAS ***
4220 INPUT"MI AZ ASZ";ED
4230 FOR I=1 TO CO:IF PRG(I)=ED THEN 4250
4235 NEXT
4240 PRINT"NEMLETEZO ASZ!":FOR L=0 TO 600:NEXT:GOTO 20
4250 PRINT"A REGI NEV";PRG$(I)
4260 INPUT"AZ UJ NEV";PPG$
4270 PRG$(I)=PPG$
4280 GOTO 20
4300 REM *** ASZ JAVITAS ***
4310 PRINT"KERESES ■■EV SZERINT,■■SZ SZERINT,VAGY    ■■ZAM SZERINT?"
4320 GET ET$:IF ET$<>"N" AND ET$<>"S" AND ET$<>"A" THEN 4320
4340 IF ET$="S" THEN GOTO 4430
4350 IF ET$="A" THEN GOTO 4490
4360 REM *** NEV SZERINT ***
4370 INPUT"A PROGRAM NEVE";RRG$
4380 FOR I=1 TO CO:IF PRG$(I)=RRG$ THEN 4400
4385 NEXT
4390 PRINT"NEM LETEZO PROGRAM NEV !":FOR L=0 TO 600:NEXT:GOTO 20
4400 PRINT"A REGI ASZ";PRG(I)
4410 INPUT"AZ UJ ASZ";ED
4420 PRG(I)=ED:GOTO 20
4430 REM *** SZAM SZERINT ***
4440 INPUT"HANYADIK ASZ";EN
4450 IF ENOCO THEN PRINT"TUL NAGY SZAM !":FOR L=0 TO 600:NEXT:GOTO 20
4460 PRINT"A REGI ASZ:";PRG(EN)
4470 INPUT"AZ UJ ASZ";PRG(EN)
4480 GOTO 20
4490 REM *** ASZ SZERINT ***
4500 INPUT"A REGI ASZ";RRG
4510 INPUT"AZ UJ ASZ";PPG
4520 FOR I=1 TO  CO
4530 IF RRG=PRG(I) THEN PRG(I)=PPG:GOTO 20
4540 NEXT
4550 PRINT"NEM LETEZO ASZ !":FOR L=0 TO 600:NEXT:GOTO 20
5000 REM *** ABC SORRENDBE RENDEZES ***
5010 PRINT"□■ABC SORRENBE RENDEZES":PRINT"■KEREM VARJON, DOLGOZOM !!!■"
5020 B=CO-2
5030 FOR I=1 TO CO-1
5040 FOR J=1 TO B
5050 IFPRG$(J)>PRG$(J+1)THENG$=PRO$(J):PRO$(J)=PRO$(J+1):PRO$(J+1)=G$:GOTO5055
5052 GOTO 5060
5055 G=PRG(J):PRG(J)=PRG(J+1):PRG(J+1)=G
5060 PRINTCO-I-1"■";B-J,"■   T"
5070 NEXT J
5080 B=B-1
5090 NEXT I
5100 GOTO20
10000 REM *** DEFINIALAS ***
10010 DIM PRO$(500),PRG(500):CO=1
10020 RETURN
```

DATORG

Külkereskedelmi Adatfeldolgozó és Szervező RT

1. Számítógépes rendszerek

– tervezése – fejlesztése – kidolgozása – adaptálása
• típusrendszerek
• egyedi adatbázison alapuló tranzakciórendszerek
• kapcsolódás központi adatbázis-szolgáltatásokhoz
• egyedi alkalmazásokat kielégítő rendszerek

2. Gépi szolgáltatások

– on-line fejlesztői környezet
– batch és interaktív rendszerek üzemeltetése
– távadatátviteli hálózat telepítése, bővítése
• terminálokkal • intelligens terminálokkal • kisszámítógépekkel
– adatgyűjtés, adatrögzítés
• lyukkártyára • mágnesszalagra • floppy diszkre
• csoportos adatgyűjtő berendezéssel
– konvertálás
• 800/1600/6250 bpi mágnesszalagról mágnesszalagra
• floppy diszkről mágnesszalagra és mágneslemezre
– gépi erőforrások bérlete

3. Tanácsadó tevékenység

– számítóközpontok telepítésénél, klimatizálásánál
– távadatfeldolgozási hálózatok kiépítésénél, bővítésénél
– gépi kompatibilitási, illesztési kérdéseknél
– hardver és szoftver termékek fejlesztésénél, adaptációjánál
– ügyvitelszervezésnél
– mikrofilm-technikát alkalmazó rendszereknél

SW-86

4. Nyomda, nyomtatványellátás

– kiadványok – propagandaanyagok – ügyviteli nyomtatványok – mikrofilm-rendszerek

Fenti szolgáltatásainkon túl is állunk szíves rendelkezésére.

DATORG 1051 Budapest, Dorottya u. 6. Telefon: 184-055 Telex: 22-5191

Klubújdonságok

A klublemezekkel kapcsolatban sok érdeklődőnek válaszolok a következőkkel. Mi ezeket és a hardvermegoldásokat nemcsak klubtagok, hanem mindenki részére értékesítjük. Klubtagok 30 százalékos kedvezménnyel kapják. Cégek is vásárolhatják. Botkormányok külön is kaphatók.

Minden további műszaki kérdéssel a szerzőt, kereskedelmi jellegű kérdéssel a 14. sz. Ügyvédi Munkaközösséget (Bp. VII., Rákóczi út 8. Ügyintéző: Ollé Tibor) keressék meg.

A legfrissebb újdonság, hogy bármilyen gyári, szerkezeti sérülés következtében véglegesen üzemképtelenné vált botkormányt a klub 1000 forintos egységáron újra cserél. A klubtagoknak járó kedvezmény természetesen itt is érvényes!

Végül ismét közlünk egy programot az első lemezről.

GETSTRING

1. BASIC-töltő

```
1 REM AZ ADATOK A $C000
  TERULETRE TOLTODNEK
5 FOR X=0 TO 142: READ A:
  POKE 12*4096+X,A: NEXT
1000 DATA169,60,133,251,
  169,3,133,252,160,0
1010 DATA169,166,145,251,
  200,192,192,208,249,160
1020 DATA0,177,251,32,210,
  255,169,157,32,210
1030 DATA255,152,72,169,
  0,32,228,255,240,251
1040 DATA170,104,168,138,
  201,13,208,6,192,0
1050 DATA240,235,208,76,
  201,20,208,22,192,0
1060 DATA240,225,136,32,
  210,255,169,166,145,251
1070 DATA32,210,255,169,
  157,32,210,255,208,207
1080 DATA196,253,240,203,
  72,165,254,208,6,169
1090 DATA58,133,250,208,
  4,169,91,133,250,104
1100 DATA24,201,48,144,2,
  176,10,201,46,240
1110 DATA10,201,32,208,172,
  240,4,197,250,176
1120 DATA166,145,251,200,32,
  210,255,76,21,192
1130 DATA145,251,169,32,32,
  210,255,169,13,32
1140 DATA210,255,96
```

2. BASIC-et használó program

```
6  REM HASZNALAT ELOTTI
RESZ
10 POKE 253,n: POKE 254,m
15 REM FORESZ
20 SYS 12*4096
30 FOR X=0 TO PEEK(782)-1:
   A$=A$+CHR$(PEEK(828+X)):
   NEXT
35 REM  A  HASZNALAT
   ELOTTI  ES  A  FORESZ
   MINDENEGYES  STRING
   ELOTT ISMETLODIK
```

Itt n a sztring maximális megengedett hossza (legfeljebb 191 karakter); ha m 0, akkor számsort, ha 1, akkor szám- és betűsort adhatunk be. A sztringet az A$ változóban tároljuk. A pufferterület a kazettapuffer. A gépi kódú program nemcsak a C-területen, hanem bárhol lehet, ahol szabad hely van, mivel a program módosítás nélkül áthelyezhető.

3. A gépi kódú rész assembler listája

```
Változók és állandók:
PLO     = $FB
PHI     = $FC
LEN     = $FD
TIP     = $FE
OUTCHR  = $FFD2
GETCHR  = $FFE4
TEMP    = $F9
TEMP2   = $FA

A sztringterület beállítása:
LDA     #$3C
STA     PLO
LDA     #3
STA     PHI

A mutató értékének nullázása,
kurzorbeállítás:
LDY     #0
LDA     #$A6
T       szubrutin címke

A kurzor értékének betöltése
a pufferterületre:
STA     (PLO),Y
INY
CPY     #$C0
BNE     T

A mutató nullázása:
LDY     #$00
START

Kurzorpozíció-kiíratás,
a kiíratópozíció
alaphelyzetbe állítása:
LDA     (PLO),Y
JSR     OUTCHR
LDA     #$9D
JSR     OUTCHR
GET

Y értékének elvermelése,
A értékének beolvasása:
TYA
PHA
LDA     $0
V

Vár a nem 00 ASCII kódra:
JSR     GETCHR
BEQ     V

A karakter ideiglenes eltárolása,
Y, majd a karakter
értékének visszavétele:
TAX
PLA
TAY
TXA

A karakter „kocsi-vissza"?
CMP     #$0D
BNE     K

Érvényes? (Ha a pufferben
még nincs karakter, akkor nem.)
CPY     $0
BEQ     GET
BNE NEX V

DEL-karakter volt?
(Ez a karaktertörlő gomb!)
CMP     #$14
BNE     N2

Érvényes?
CPY     $0
BEQ     GET

A mutató pozícióértékének
csökkentése,
a DEL-karakter kiírása:
DEY
JSR     OUTCHR

A kurzor új pozíciójának
eltárolása és kiírása:
LDA     #$A6
STA     (PLO),Y
JSR     OUTCHR

A kiíratási pozíció alaphelyzetbe
állítása a képernyőn:
LDA     #$9D
JSR     OUTCHR

A következő karakter
beolvasása jön:
BNE     GET
N2

Van még hely a pufferben?
CPY     LEN
BEQ     GET

A karakter ideiglenes tárolása,
típusának meghatározása
(szám, vagy szám és betű),
az érvényességi skála beállítása:
PHA
LDA     TIP
BNE     A
LDA     #$3A
STA     TEMP2
BNE     S
A

A szám- és betűtípus beállítása:
LDA     #$5B
STA     TEMP2
S

A karakter elővétele a veremből:
PLA

A CARRY-bit kikapcsolása,
a karakter elfogadhatóságának
vizsgálata:
CLC
CMP     #$30
BCC     C
BCS     U
C

Tizedespont?
CMP     #$2E
BEQ     P

Üres hely?
CMP     #$20
BNE     GET
BEQ     P
U

A skálán kívül van a karakter?
CMP     TEMP2
BCS     GET
P

A karakter eltárolása,
a mutató továbbléptetése,
a karakter kiíratása,
a karakterbeolvasás kezdetéhez
visszatérés:
STA     (PLO),Y
INY
JSR     OUTCHR
JMP     START
NEX V

A kocsi-vissza karakter eltárolása,
üres hely kiíratás,
kocsi-vissza kiíratás,
vissza a BASIC-be:
STA     (PLO),Y
LDA     #$20
JSR     OUTCHR
LDA     #$0D
JSR     OUTCHR
RTS
```

DR. SIMONYI ENDRE

UTCÁN ÁT...

ECONORG Szaküzlet
Budapest, VI., Szinyei Merse u. 1. Telefon: 127-628
Nyitva: hétfő, kedd, szerda, péntek: 9–17 óra,
csütörtök: 10–20 óra

Új üzlettel gyarapodott a számítástechnikai boltok hálózata Budapesten, az ECONORG Számítástechnikai Közös Vállalat (volt VSZFT) jóvoltából. Ez év elején döntöttek úgy, hogy olyan szaküzletet hoznak létre, amely mind a nagyközönség, mind a hivatásos felhasználók igényeit kielégíti. Így nyílt meg július 1-én ez a bolt.

Fő profiljuk egyrészt saját fejlesztési programjaik forgalmazása (bemutatás, tanácsadás) R22, TPA 11/48-ra, illetve mikrogépekre, másrészt a hagyományos bizományosi tevékenység. Az egyedi programkidolgozási igényeket az ECONORG fejlesztőgárdája elégíti ki, elsősorban hagyományos alkalmazási területeken, úgymint mezőgazdaság, ipar, külkereskedelem és vezetői információs rendszerek. A bizományosi tevékenység keretében számítógépeket, alkatrészeket, floppykat, szoftvereket stb. vesznek be.

Vállalják vállalatok, kisszövetkezetek, gmk-k, magánszemélyek hardver- és szoftvertermékeinek bemutatását is. A szoftverbemutatókat és a vásárlást a helyszínen felállított C64 és Floppymat SP gépek segítik.

Barátságos SYSTER-szoftverek

MTA — SZTAKI COSY MŰSZAKI FEJLESZTŐ LEÁNYVÁLLALAT

Közelebb, még közelebb ... sőt ... közel! Ne legyen távolság térben és időben a FELADAT és FELHASZNÁLÓ között! Ez a mottó, melyet ma mindenki hangoztat, aki csinálja a szoftvert s aki segíteni akar a problémák megoldásában.

Mert hogy is volt régen? Régen ...? Sokszor még ma is. A tisztelt felhasználó (általában egy szorgalmas beosztott, sőt helyette – sokszor – egy tökéletesítés megszállottja: akit a vállalatnál többen, enyhén szólva, egy kicsit furának tartottak) *kitalálta: mit* kellene számítógépen kiszámolni, nyilvántartani, feldolgozni, gépre vinni stb. Majd *elment: meggyőzni* a főnökét arról, hogy tényleg ki kellene számolni azt, amit kitalált. Tegyük fel, ez nem tartott tovább egy hétnél. Az iroda vagy osztály vezetője kulturált szemléletű és jó vezető volt. Íratott egy feljegyzést. Mikor elkészült ... legalább egyszer átjavította, újragépeltette ...: *elküldte följebb*. A főosztályvezetőtől mondjuk két hét alatt visszajött a válasz: *ki kell deríteni: hol, mennyiért, mennyi idő alatt* számolható ki (dolgozható fel ...) a feladat. Az osztályvezető a legtöbbször elküldte azt aki kitalálta, vagy a tökéletesítés megszállottját, esetleg egy fiatalt – aki nemrég végezte az egyetemet. S kiderült, hogy pl. a hatvanas évek végén Pesten, legkevesebb 50 ezer forintért, legalább *egy hónap alatt megoldható egyetlen számítás*. S mindezt úgy két hét *nyomozói munkával* lehet kideríteni. De még nincs vége. Most írni kellett a vállalat igazgatójának, hogy *engedélyezze* mindezt: a legjobb esetben is eltartott ez a procedúra is vagy egy hónapig. Aztán a *szerződést is meg kellett kötni* a szervező vállalattal: még egy hónap. Szóval *úgy 4–5 hónap alatt* elkészíthető volt *egyetlen* számítás. S közben minden ment flottul, de mi lett volna a helyzet, ha bárhol elakad. S a legjobb szervezést feltételezve az előkészítési időszak alatt már folyt – szerződés nélkül – a munka, az adatok, szövegek előkészítése.

S jól emlékeznek sokan: általában egy féléves munka az egyhónapos helyett belekerült gépidővel együtt 300–700 ezer forintba. S ne is beszéljünk a még ma is évekig húzódó többmilliós ügyviteli adatfeldolgozó munkálatok előkészületeiről. Az első eredményt öt év múlva, ha lehetett látni.

Az idők folyamán sok intézmény arra esküdött, hogy a hosszú átfutási idő alapvető oka a külső szervezők-számítástechnikusok hiányos hely- és szakismerete. Ezért a számológépek „nagy" gépei és óriási légkondicionált termei a vállalat szervezetébe épültek be. Sajnos közben a jó szakemberekből az igények növekedésével egyre kevesebb lett. Az átképzősök igen színvonalasan, de csak részlegesen pótolták a hiányt. A vállalati rendszeres adatfeldolgozás zöme megvalósult. De az újabb feladatok megoldását a számítógépes beruházásokat is korlátozó szűkös keretek tovább akadályozták. Hiszen már az új számítógép üzembe helyezésekor is épp, hogy elég volt a bővített gépkapacitás. No meg sokba is került az egyszeri nagyberuházás.

Ismeretes azonban, hogy a szakemberhiány a számítástechnikai alkalmazásban fejlettebb területeken is gond. Sokan pozitívan emlegetik, hogy milyen jó, hogy mi nekik szakembert – programozót – munkát exportálunk, s közben mennyire fejlődik a számítástechnikai kultúránk. Szerintem, sajnos ez tovább rontja a hazai szakember-ellátottságunkat. S lehet, hogy sokkal hatékonyabb lenne az itthoni alkalmazás exportnövelő befolyása, ha hazai vizeken hajózna minden – tehetséges – programozó.

A hiányt először a 60-as évek közepén a WANG-cég próbálta betölteni asztali számítógépével. Közben azonban elterjedtek az időmegosztásban dolgozó terminálok. Egy nagy gépen több felhasználó a saját szobájában dolgozott a terminálokon. Csökkent a programlyukasztók száma – de lassulni kezdtek a feldolgozások. Mindenki egyszerre akart dolgozni, a gép mint díszpécser ha megfeszült sem tudott gyorsítani. A szükséges alapszoftveresek időlegesen segíthettek volna, ha elegen vállalták volna a piacra dolgozás helyett: a hálátlan labor-orvosi feladatot a szülész-orvosi helyett.

Megjelentek az érdekeltség ösztönözte GM-ek. S bár ők megint külső szakemberek, időlegesen enyhítettek a helyzeten. Az igazi segítséget azonban a személyi és professzionális asztali számítógépek jelentik mintegy 1980 óta. Ezek elérhetően olcsóak és ... egy *új szoftver-szemléletet* is hoztak.

Az új szemlélet a közvetítők hiányában rejlik. Nem kell a beosztott mérnöknek, közgazdásznak, gépírónőnek, ügyintézőnek vagy díszpécsernek ... tervezőnek, tanárnak avagy vezetőnek a maga számításaival, gépelésével vagy nyilvántartásával egy engedélyezési-szervezési-magyarázkodási procedúrát – vagyis egy kálváriát – végigjárni-kínlódni. „Beül" a gépébe és megírja a levelét. Elrontotta? Nem baj: a gép tárolta, és csak a hibát kell javítani – nem újragépeltetni az amúgy is szűkkapacitású *gépírónői* állomány csinosabbjával. Levelezhet úgy is, hogy a címzettről az összes nyilvántartandót: és ezek között azt is, hogy mi teljes megnevezése, a helység ahol megtalálható, az irányítószáma, utcaneve, házszáma, az igazgató neve avagy az ügyintézője – mind-mind egymás mellett gépen tárolja. Ezek közül azt gyűjt ki, akiknek éppen most kell levelet írni. S akár több ezer levelet összesen 8–10 gombnyomással automatikusan elkészíthet professzionális számítógépünkön – a *SYSTER-en*. S közben használhatja a billentyűzeten, a képernyőn és a nyomtatón a *teljes magyar ékezetes karakterkészletet*.

Ha pénzügyi, forgalmi avagy ártervet készített, nagy kockás ragasztott lapokra írta, majd radírral javította-maszatolta, kézzel gyűrte a terveit. Ma 20–30 kezelési szabály megtanulása után 1–2 nap után órák alatt számolhatja ki napok munkáját. A táblázaton úgy gondolkodhat, hogy mondjuk azt mondja: legyen a harmadik és hatodik oszlop szorzata a nyolcadik oszlop, vagy csinálj részösszegeket és végösszesent itt és itt. A táblázat-számító programunk ezt megteszi; sőt tárolható, javítható, nyomtatható és ... beépíthető egy szövegfeldolgozó programunkkal készített jelentésbe is.

Azért szemléltetjük az új szoftver-szemléletet a SYSTER-en, az MTA SZTAKI által kifejlesztett magyar professzionális és mégis igen olcsó (sőt legolcsóbban liesingelt) számítógépen, mert ez szinte az egyetlen asztali gép, mely *minden magyar betűt mindenütt* ismer. Továbbá mert számunkra elfogadott tény, hogy nem szeretik az emberek az általánosságokat. Konkrétan meg csak egyet – akkor legalább egyet a legjobbak közül – ismertethetünk.

A SYSTER azonban a hagyományos szemléletű szoftverekkel is el van látva. Hiszen tudjuk, hogy a célra készített rendszerek gyorsabbak lehetnek. Bár elkészítési idejük jóval lassabb. Mindenesetre minden Magyarországon használt programnyelv rendelkezésre áll: BASIC interpreter és fordító, PASCAL, FORTRAN, C stb. van programszerkesztő, diszkjavító és egyéb segéd-szoftver is. Aki meg már kész rendszerekre vágyik, annak áruforgalmi, költségelszámoló, termelésirányító nagy rendszerek állnak rendelkezésére.

Maga a SYSTER bár 8-bites, de 128K-ra bővíthető. (A második 64K memória-diszkként gyors háttérmemóriaként használható). Két-négy darab egyenként formázottan 700K-s lemezegység, sőt 10–27M-es WINCHESTER is hozzákapcsolható. Több száz SYSTER csatolható 1000 méteres távolságban hálózatba. Sőt bármilyen Magyarországon található számítógéphez is csatolható, terminálként programozás nélkül. A SYSTER a PRIMO nővére, és a PRIMO-k hálózati csatolójaként azokról adatokat fogad a SYSTER-diszkekre (és visszafelé is áramoltatni képes innen az információt).

Azt mondhatjuk tehát, hogy a SYSTER, a PRIMO és a LANPBOX, valamint a felhasználó által közvetlenül, szervezők és programozók nélkül használható szoftverekkel az MTA SZTAKI COSY Műszaki Fejlesztő Leányvállalat egy teljesen barátságos, igen olcsó komplex rendszerrel alátámasztja a számítástechnika fejlődéséről vázolt képünket. S e cikkel nem is volt más szándékunk.

LÉPÉSRŐL LÉPÉSRE

Sorozatunk előző részében azt kezdtük vizsgálni, hogy milyen eljárások, stratégiák segítségével sakkozik az ember. Láttuk: az ember egyáltalán nem úgy elemez egy sakkállást, hogy nagy számban áttekint lehetséges lépéssorozatokat, és az ezek után létrejött állásokat kiértékeli. Az ember már első pillanatban, amikor ránéz egy sakkállásra, nem a figurák konkrét helyét érzékeli, hanem a figurák között fennálló, sakkszempontból értelmes kapcsolatokat, relációkat, lehetséges játékeszméket. Ha ez nem így lenne, akkor egy értelmetlen állást ugyanolyan jól (vagy ugyanolyan rosszul) tudna az ember rövid idő alatt megjegyezni, mint egy értelmeset.

Ezzel az emberi gondolkodásmódok kutatásának egyik legérdekesebb területébe ütköztünk: amit az előbb mondtunk, az nemcsak a sakkozó gondolkodására igaz, hanem szinte minden gondolkodási tevékenységére. Az ember egy olyan szituációt, amely számára akár csak egy kicsit is ismerős korábbi tapasztalataiból, úgy észlel és úgy elemez, hogy közben folyamatosan korábban kialakult mintáira, sémáira támaszkodik, és eközben maguk a sémák is módosulnak, megváltoznak, sőt szaporodnak. Az ilyen gondolkodási sémák vizsgálata a modern kognitív pszichológia egyik legerősebben kutatott területe. Magyarul Ulric Neisser Megismerés és valóság című könyvében olvasható kiváló összefoglalás e terület jelenlegi állásáról.

Visszatérve a sakkhoz, az ember mintái, sémái segítségével egy adott állásból többnyire konkrét variációk pontos kiszámítása nélkül tud kiértékelni hosszabb távú fejleményeket. Valószínűleg külön mintái vannak annak eldöntésére, hogy melyek azok az állások, amelyek ténylegesen sokágú, hosszú és pontos konkrét számításokat igényelnek.

Ha elfogadjuk, hogy az ember mintái, sémái segítségével észlel és elemez egy sakkállást, akkor felmerül az a kérdés, hogy hány ilyenfajta sémája lehet. A kérdés egyfelől nagyon természetes, másfelől mégis abszurdnak tűnik: van egy homályos fogalmunk, a séma; azt nem tudtuk megmondani, hogy pontosan mit is értsünk e fogalmon, de máris arra vagyunk kíváncsiak, hogy hányan lehetnek. Érdekes módon mégis ez a látszólag teljesen megalapozatlan kvantitatív kérdés bizonyult kezelhetőbbnek, méghozzá éppen a sakkozók gondolkodásának vizsgálata kapcsán.

A következőkben röviden körvonalazzuk a közgazdasági Nobeldíjas Herbert Simon és munkatársai módszerét, amelynek segítségével a sémák darabszámára mennyiségi becslést tudtak adni. Látni fogjuk, hogy az eredmény a sakkprogramozás fejlődése szempontjából is iránymutató jelentőségű.

A sémák egyelőre meglehetősen homályos fogalmát úgy tesszük a mennyiségi becslés számára megfoghatóvá, hogy feltételezzük: az embernek van egy rövid idejű memóriája, ahol hosszabb ideig ugyan nem képes információt tárolni, de ideiglenesen, csak elemzés céljaira igen. E rövid idejű memória létezését sokféle kísérlettel igazolták; egyelőre higgyük ezt el, és következő tulajdonságait is tekintsük igazoltnak:

1. Az ember a rövid idejű memóriájában csak olyan információt képes megragadni, amely kapcsolódik a hosszú idejű memóriájában egy más meglévő sémához.

2. Az ember rövid idejű memóriájának kapacitása nagyon korlátozott: csak mintegy hét információegységet képes benne egyidejűleg tárolni.

3. A rövid ideig megmutatott sakkállások esetében az az idő, amit az állás megtekintésére adtak, nem elégséges arra, hogy az ember ezalatt rövid idejű memóriájából egy információegységet hosszú idejű memóriájába továbbítson.

Vegyük észre, hogy a sémák fogalma csak az 1. pontban szerepel. Mégis, ennyi már elegendő ahhoz, hogy az így „megfogott" sémák darabszámára mennyiségi becslést adjunk. A sakkállásokat rekonstruáló kísérletben ugyanis azt tapasztalták, hogy a nagymesterek és csakis ők voltak képesek arra, hogy egy mesterjátszmából való állást gyakorlatilag száz százalékos biztonsággal hibátlanul rekonstruáljanak. A nagymesterek is csak mesterjátszmából vett középjátékbeli állások hibátlan rekonstrukciójára voltak képesek; egy amatőr parti középjáték-állásánál nemegyszer hibáztak.

Most fordítsunk egyet a dolgon: ezek szerint a nagymesterek hosszú idejű memóriájában tárolva kell lennie legalább annyi sémának, hogy közülük legfeljebb annyival, mint amennyi a rövid idejű memória kapacitása, bármelyik, mesterjátszmában létrejöhető sakkállás egyértelműen leírható legyen. Hiszen a nagymester képes csak a rövid idejű memóriájának használatával bármelyik ilyen állást gyakorlatilag hibátlanul rekonstruálni.

Ennek a gondolatmenetnek a segítségével tehát – ha megbecsüljük, hogy egyáltalán hányféle állás fordulhat elő egy mesterjátszmában – alsó becslést kaphatunk arra, hogy hányféle sémának kell minimálisan lennie egy nagymester fejében. A számítás technikai részleteit nem ismertetjük, mindenesetre az így leírt gondolatmenet alapján reálisnak látszik, ha a nagymesterek által ismert sémák számát 10 000 és 100 000 közé becsüljük.

Ezek után a rekonstruálás során elkövetett hibák alapján visszaszámolva becslést kaphatunk arra is, hogy a nemzetközi nagymesternél gyengébb játékerejű játékosok körülbelül hány, sakkal kapcsolatos sémát ismerhetnek. A csökkenés mértéke megdöbbentő: egy mesterjelölt erejű játékos valószínűleg már csak legfeljebb egy-kétezer sémát ismer, egy első osztályú játékos pedig csak néhány százat.

Az amatőrök és a nagymesterek sakkjátéka között mindenki számára érzékelhetően óriási minőségi különbség van: egy nagymester általában képes egyszerre 20–25 erős mesterjelölttel szimultánt játszani, és többnyire igen jó, 90 százalék körüli eredményt ér el. Ezt a hatalmas minőségi különbséget eredményeink váratlan szemléletességgel teszik érthetővé: a tudáskülönbség az ismert minták számában több mint egy nagyságrendnyi különbséggel jár együtt. Így tehát egy természetes mérték segítségével is tudunk válaszolni egy olyan, amúgy igen nehezen megfogható kérdésre, hogy „mennyivel erősebb egy 2600-as Élő-pontszámú játékos egy 2300-asnál".

A nagymesteri játékerőhöz szükséges több tízezernyi sémát az ember hosszú és intenzív tanulás és gyakorlás során szerzi meg. Ez a folyamat minimum tízéves, napi sokórás elfoglaltságot igényel. Ez alól még az ismert csodagyerekek (Capablanca, Reshevsky, Fisher) sem voltak kivételek. Ráadásul a sémák többsége olyan, hogy nem tudjuk őket szavakban kifejezni: a legjobb tankönyvek is csak azt tudják elősegíteni, hogy a tanulók ráérezzenek a példákban rejlő általános sémákra. Valószínű, hogy a sémák egymásra épülve, egymást kiegészítve, a tanulás és gyakorlás során folyamatos változás és finomodás közben alakulnak ki.

Ha valaki egy sakkprogram készítésénél megpróbálja azt az utat választani, hogy utánozza azt a módot, ahogyan az ember sakkozik, az eddigiek ismeretében ijesztően nagy fába vágja a fejszéjét. Hiszen egyfelől kitűnik, hogy sok tízezerféle sémát kell algoritmikusan megfogalmaznia, beprogramoznia és főleg egymással összeszerveznie ahhoz, hogy az emberi csúcsteljesítményt megközelíthessé. Másfelől még azt is nehéz megmondania, hogy egyáltalán mit tekinthet sémának, melyek azok a sémák, amiket egy nagymester automatikusan meglát. Némelyik egyszerű: „figyelj az alapsor gyengeségére", a legtöbb séma azonban nagyon intuitív, még szavakba is nehezen foglalható. A másik, és talán még nehezebb probléma a sémák illesztése a konkrét sakkálláshoz. Ezt a feladatot az ember „ráérzéssel" végzi el. Azok a technikák, amelyekkel az ilyesmit már sikerült számítógépre adaptálni, nagyon speciálisak, és mondhatnánk, kezdetlegesek.

Néhány egyszerűbb séma, mint a korábbiakban láttuk, a jelenlegi sakkprogramok alapját alkotó minimax algoritmus kereteibe is beilleszthető, és valóban nagymértékben képes a program játékerejét javítani. Ezekkel a technikákkal azonban legfeljebb néhány tucat séma beépítéséről lehet szó, láttuk, hogy ebben a keretben a sémák számának emelése áttekinthetetlen bonyodalmakhoz vezetett, és egy idő után a hatékonyságot kifejezetten csökkentette.

Olyan komplett sakkprogram, amely alapjában mintaillesztéssel dolgozik, és legfeljebb ezt egészíti ki valamiféle keresési mechanizmus, eddig még nem készült. Néhány részfeladatra azonban, például bizonyos típusú végjátékokra vagy speciális kombinációk megtalálására készültek már ilyen programok. Ezek az ember számára néha elképesztően bonyolult kombinációkat is felfedeznek, nem egy esetben klasszikus tankönyvpéldákról kialakult, több évtizedes véleményeket változtattak meg, ugyanakkor előfordul, hogy az ember számára evidens játékmódokat nem találnak meg.

Az elkészült séma elvű sakkprogramokba (vagy mondjuk inkább: programrészekbe) még mindig csak néhány száz sémát sikerült általában beépíteni; igen valószínű, hogy viszonylagos és néha szinte érthetetlen gyengeségük ebből fakad. Nagyon finom sémakészletük van arról a részterületről, amiben specialisták, de a sakk általános elveiről nem tudnak eleget. Nagymesteri erővel játsszák a bonyolult, ellenkező színű futós végjátékot, és utána a kialakult egyszerű vezérvégjátékban óriásit melléfognak.

Nehéz itt prognózist adni: már körvonalazódik a sakkprogramok egy új generációja, de még beláthatatlanul sok problémát kell ezen az úton megoldani. Teljesítményük egyelőre még meg sem közelíti a régi elvű, évtizedek alatt kicsiszolódott programok teljesítményét, de a régi típusú sakkprogramok teljesítőképességének határai egyre inkább kirajzolódnak. Érdekes módon a mesterséges intelligencia legtöbb más területén is nagyon hasonló a helyzet. Sorozatunk következő, befejező részében ebbe az irányba mutatunk némi kitekintést.

MÉRŐ LÁSZLÓ

Primóra

KÉT REFLEXJÁTÉK

A *Vadászat*ban a képernyőn kereszül futó vadra tetszőleges billentyű lenyomásával lehet lőni.

Kerüld ki az akadályokat, vagyis a felfelé haladó Ø-kat az oldalirányban vezérelhető ponttal! Ennek a programnak az az érdekessége, hogy a PRINT$ működését is bemutatja.

SOMOGYI GYÖRGY

```
2 REM Vadászat
3 FOR Z=1 TO 25
4 P=1Ø+RND(1Ø)
5 A=1Ø+RND(7)
8 FOR K=1 TO 19
1Ø PRINT CHR$(12)
15 FOR Q=1 TO P
2Ø SET(K*1Ø+Q,A)
25 NEXT Q
3Ø B$=INKEY$
35 IF C>Ø.5 THEN 55
4Ø IF B$<>"" C=1
5Ø IF C<Ø.5 THEN 9Ø
55 G=POINT(12Ø,C*1Ø)
6Ø SET(12Ø,C*1Ø)
7Ø C=C+1
8Ø IF G=Ø THEN 9Ø
81 TA=TA+1
82 PRINT CHR$(7)
84 PRINT 1Ø,1,TA;"találat";Z;"menetből."
86 FOR I=1 TO 2ØØØ
88 NEXT I
9Ø SET(12Ø,1)
1ØØ NEXT K
1Ø5 C=Ø
1Ø6 FOR I=1 TO 5ØØ
1Ø7 NEXT I
11Ø NEXT Z
12Ø PRINT "A TALÁLATOK SZÁMA=";TA
```

```
2Ø REM Kerüld ki az akadályokat
25 DIM JJ(15)
3Ø PRINT "A PONTOT BALRA AZ a BETŰ,
   JOBBRA AZ ó BETŰIRÁNYÍTJA.":
   PRINT "Írj be egy betűt!" : INPUT A$
32 REM a 3Ø-as sorban a betűirányítja
   nem véletlenül van egybeírva
35 PRINT CHR$(12)
4Ø Z=1Ø
5Ø FOR I=1 TO 199
52 FOR E=1 TO 15Ø
53 REM Ez a ciklus a sebességet állítja be
54 NEXT E
6Ø A=RND(15)
7Ø PRINT$ 15,1,":                :"
75 REM A 7Ø-es sorban 16 db betűköz van
8Ø PRINT$ 15,A+2,"O"
9Ø W$=INKEY$
1ØØ IF W$="A" OR W$="a" Z=Z-1
11Ø IF W$="ó" Z=Z+1
115 PRINT$ 6,Z,"."
12Ø IF Z>18 OR Z<1 THEN 2ØØ
121 JJ(1)=A
122 FOR L=6 TO 1 STEP -1
123 JJ(L)=JJ(L-1)
124 NEXT L
125 AJ=JJ(6)
13Ø IF Z<>AJ+2 OR I<4 THEN 19Ø
135 K=K+1
14Ø PRINT$ 6,Z,"*              ";K;" sérült!"
15Ø PRINT CHR$(7)
19Ø NEXT I
192 PRINT CHR$(12);K;" sérült!"
    :K=Ø:GOTO 2Ø1
2ØØ PRINT "Letértél az útról!"
2Ø1 END
```

LASER–210-re

ŰRUTAZÁS

Űrrepülőgéppel idegen objektum felett kell repülni úgy, hogy a szembejövő csillagoknak ne ütközzünk. A gépet a . és a SPACE billentyűkkel, valamint botkormánnyal lehet irányítani. Haladás közben fogy az energia, amit pótolni a ■ alakzatokból lehet.

```
5 FORI=ØTO43:READA:
  POKE3ØØØØ+I,A:
  NEXT:POKE3Ø862,48:
  POKE3Ø863,117
1Ø CLS:S=28672:POKE3Ø744,Ø:Y=S+1
74:F=5Ø
15 PRINT@2,"ENERGY:":PRINT@15,
  "POINTS:"
2Ø REM
3Ø X=RND(13)*32:IF E/2=INT(E/2)
  THEN POKES+X,1Ø6
35 IF E/28=INT(E/28)
  THENPOKES+X,99
4Ø E=E+1:IFE/14=INT(E/14)THEN
  GOSUB 4ØØ
5Ø FORI=S+48ØTO48Ø
  -(RND14)*32)+S
  STEP-32:POKEI,34:NEXTI
6Ø A=INP(43)AND31:POKEY,96
7Ø IF A=3Ø OR INKEY$="."
  THEN Y=Y-32:F=F-1
8Ø IF A=29 OR INKEY$=" "
  THEN Y=Y+32:F=F+1
9Ø IF PEEK(Y)=34 THEN GOSUB 3ØØ
1ØØ POKEY,95
11Ø IF PEEK(Y-1)=1Ø6 THEN GOSUB
  2ØØ
115 IF PEEK(Y-1)=99 THEN GOSUB
  7ØØ
12Ø O=PEEK(Y-1):IFO=159ORO=175OR
  O=191ORO=2Ø7THEN GOSUB 6ØØ
125 POKEY,96
13Ø Q=USR(P)
131 POKEY,95
135 IF PEEK(Y-32)=34THEN GOSUB
  3ØØ
14Ø F=F-1:IF F<1THEN GOSUB 5ØØ
15Ø PRINT@1Ø,F:PRINT@23,E:GOTO2Ø
2ØØ SOUND1Ø,1:R=R+1
21Ø IFR=5THENI4ØØ
22Ø PRINT@28,R
23Ø RETURN
3ØØ PRINT"BECSAPODOTT":
  SOUND31,1:3Ø,2:GOTO1ØØØ
4ØØ POKES+X,(16*RND(3)+143):RETURN
```

```
5ØØ SOUND3Ø,1;2Ø,1;1Ø,1:GOTO1ØØØ
6ØØ SOUND3Ø,1
61Ø IFO=159THENF=F+15
62Ø IFO=175THENF=F+25
63Ø IFO=191THENF=F+3Ø
64Ø IFO=2Ø7THENF=F+2Ø
65Ø RETURN
7ØØ SOUND15,1:W=RND(5Ø)-15
71Ø E=E+W
72Ø RETURN
8ØØ DATA33,255,113,62,224,94,35,115,43,
   43,61,32,248,255,94,35
81Ø DATA115,43,43,61,32,248,33,2,54,1Ø5,
   54,Ø,62,16,1,32,Ø,33,Ø
82Ø DATA112,Ø,54,96,9,61,32,25Ø,2Ø1,Ø
1ØØØ CLS
1Ø1Ø PRINT"ON ELVESZTETTE A JATEKOT"
1Ø2Ø PRINT"EDDIG"E"PONTOT ERT EL"
1Ø3Ø RESTORE
1Ø4Ø IFPEEK(26624)<>255THENE=Ø:
   GOTO 1Ø ELSE 1Ø4Ø
1Ø5Ø END
```

A program minden előrelépésnél ad egy pontot, de a # alakból -15-től 35 pontig lehet szerezni. Vigyázni kell arra is, hogy a repülőgép az alsó karakterekbe ne csapódjon be. Jó játékot!

ZX-Spectrumra

TORPEDÓ

A program a közismert torpedó-játék számítógépes változata. Igaz, hogy kicsit hosszúra kerekedett, de sikerült a gépet annyira „kiokosítani", hogy igazán egyenrangú partner, sőt sokszor még szerencsésebb is a játékosnál.

A játék lényege, hogy egy 10 × 10-es mezőben el kell helyezni 1 db hármas, 2 db kettes és 3 db egyes hajót. A hármas hajó mindig L alakú, de el lehet forgatni, akárcsak a kettes hajókat, viszont nem lóghatnak ki a játékmezőből, és nem érhetnek össze. Csalás, tévesztés kizárva. Előbb a gép helyezi el a figuráit, de hogy hova, azt csak a játék végén mutatja meg. A játékos adott menüből gyorsan építheti fel mezejét a típus és a ponttal jelölt rész koordinátáinak meghatározásával.

A játékos lő először, sor- és oszlopmeghatározással. Ha nem talált, a gép mezejében az adott koordináták által meghatározott területen megjelenik egy pont, találat esetén sakktáblaszerű karakter és a dialóg mezőben a „TALÁLT" vagy „ELSÜLLYEDT" üzenet. „TALÁLT" esetén természetesen addig kell majd körbelőni az adott területet, amíg el nem süllyed a hajó.

A gép a szemléletesség kedvéért kétféle karakterrel jelzi lövését. Ha olyan helyre talál, ahol nincs a játékosnak hajója, akkor egy villogó kérdőjelet, ha viszont talál, akkor villogó csillagot jelenít meg az adott koordinátában. Ez a tulajdonsága a programnak csak a játékos válaszadását segíti elő. Gombnyomással kell a lövést kiértékelni, de csalni nem érdemes, mert akkor a gép abbahagyja a játékot. A játékmező felállítását és a célkoordináták meghatározását a véletlengenerátor segítségével végzi, így egyenrangú ellenfélnek tekinthető.

```
   1 REM ***********************
   2 REM * TORPEDO   ©Bogos Z.*
   3 REM ***********************
   4 DATA 0,0,0,BIN 00011000,BIN
 00011000,0,0,0: FOR i=0 TO 7: R
EAD j: POKE USR "b"+i,j: NEXT i
   5 DATA 128,128,128,128,128,12
8,128,255: FOR i=0 TO 7: READ j:
 POKE USR "a"+i,j: NEXT i
   6 DEF FN m(x,y)=x-INT (x/y)*y
 DEF FN a(x,y)=(x<1)+(x>10)+(y<
1)+(y>10)
   7 LET u=BIN 10101010: LET v=B
IN 01010101: FOR i=0 TO 7: LET j
=FN m(i,2): POKE USR "c"+i,u*(j=
0)+v*(j<>0): NEXT i
  10 DIM s(6): DIM g(10,10): DIM
 j(10,10): DIM w(10,10)
  20 RANDOMIZE
  30 LET elso=0: LET taj=0: LET
tag=0: LET kerul=0
  35 CLS : PRINT AT 10,4;"FELALL
ITOM SAJAT MEZOMET"
  40 FOR i=1 TO 10: FOR j=1 TO 1
0: LET g(i,j)=0: LET j(i,j)=0: L
ET w(i,j)=0: NEXT j: NEXT i
  50 FOR i=1 TO 6: LET m(i)=INT
(RND*100): IF i=1 THEN GO TO 130
  60 FOR j=1 TO i-1: LET tav=ABS
 (m(j)-m(i))
  70 IF tav=0 OR tav=1 OR tav=9
OR tav=10 OR tav=11 THEN LET i=i
-1: GO TO 130
 110 NEXT j
 130 NEXT i
 140 FOR i=1 TO 6
 150 GO SUB 5000
 160 LET g(a,b)=1
 165 NEXT i
 170 LET i=1
 175 GO SUB 5000
 177 LET s11=a: LET s21=b+1: LET
 s12=a-1: LET s22=b: GO SUB 400
 180 IF i=1 THEN LET s21=b-1: GO
 SUB 400
 190 IF i=1 THEN LET s12=a+1: GO
 SUB 400
 200 IF i=1 THEN LET s21=b+1: GO
 SUB 400
 210 IF i=1 THEN GO TO 50
 215 GO SUB 5000
 220 IF i=2 THEN LET s1=a: LET s
2=b+1: GO SUB 640
 230 IF i=2 THEN LET s1=a+1: LET
 s2=b: GO SUB 640
 240 IF i=2 THEN LET s1=a-1: LET
 s2=b: GO SUB 640
 250 IF i=2 THEN LET s1=a: LET s
2=b-1: GO SUB 640
 260 IF i=2 THEN GO TO 50
 265 GO SUB 5000
 270 IF i=3 THEN LET s1=a+1: LET
 s2=b: GO SUB 640
 280 IF i=3 THEN LET s1=a: LET s
2=b+1: GO SUB 640
 290 IF i=3 THEN LET s1=a-1: LET
 s2=b: GO SUB 640
 300 IF i=3 THEN LET s1=a: LET s
2=b-1: GO SUB 640
 310 IF i=3 THEN GO TO 50
 320 FOR i=1 TO 10: FOR j=1 TO 1
0
 340 LET w(i,j)=g(i,j)
 350 NEXT j: NEXT i
 360 GO TO 1047
 400 LET rr=0: LET s1=s11: LET s
2=s21: GO SUB 640: IF el=0 THEN
RETURN
 430 LET rr=1: LET s1=s12: LET s
2=s22: GO SUB 640: IF el=1 THEN
LET g(s11,s21)=1
 440 RETURN
 640 LET r=(s1+s2)-(a+b): GO SUB
 880
 650 IF rr=0 THEN RETURN
 660 IF el=1 THEN LET g(s1,s2)=1
 690 RETURN
 880 LET el=0: IF FN a(s1,s2)>0
THEN RETURN
 900 RESTORE 910
 910 DATA -1,1,-1,2,0,2,1,2,1,1
 920 FOR c=1 TO 5: IF s2=b THEN
GO TO 960
 940 READ u,v: LET u=a+u: LET v=
b+v*r
 950 GO TO 970
 960 READ v,u: LET u=a+r*u: LET
v=b+v
 970 IF FN a(u,v)>0 THEN GO TO 1
000
 990 IF g(u,v)=1 THEN RETURN
1000 NEXT c
1010 IF rr=1 THEN LET i=i+1
1020 LET el=1: RETURN
1040 CLS : FOR i=1 TO 10: FOR j=
1 TO 10: IF w(i,j)=1 THEN PRINT
"█";
1043 IF w(i,j)=0 THEN PRINT "·";
1044 NEXT j: PRINT : NEXT i
1046 RETURN
1047 CLS : LET a=4: LET b=32
1048 PRINT AT 2,6;"JATEKOS
   GEP"
1050 FOR i=1 TO 2: FOR j=0 TO 10
1051 PLOT b+j*8,56: DRAW 0,79: I
F j=10 THEN GO TO 1060
1053 PRINT AT 3,j+a;j+1
1060 NEXT j: LET a=16: LET b=144
 NEXT i
1061 LET a=32: LET b=1
1062 FOR i=0 TO 10: PLOT 223,56+
8*i: DRAW -79,0: IF i=10 THEN GO
 TO 1065
1064 PRINT AT 14-i,16;10-i
1065 PLOT 111,56+8*i: DRAW -79,0
: IF i=10 THEN GO TO 1068
1067 PRINT AT 14-i,2;10-i
1068 NEXT i
1069 PRINT AT 17,2;"█                      █
 ██  █ █"
1070 PRINT AT 18,2;"█ ██          █
 ██"
1071 PRINT AT 20,2;"1     2                1
   2     3     4"
1080 INPUT "A harmas hajo helyze
te: ";poz: INPUT "Koordinatak: s
or: ";i;" oszlop: ";j
1085 LET tip=3: GO SUB 5040
1095 IF hiba=1 THEN GO TO 1080
1100 LET j(i,j)=2: PRINT AT i+4,
j+3;"█": IF poz<>1 THEN GO TO 11
15
1105 LET j(i-1,j)=2: PRINT AT i+
3,j+3;"█": PRINT AT i+4,j+4;"█":
 LET j(i,j+1)=2
1110 GO TO 1145
1115 IF poz<>2 THEN GO TO 1130
1120 LET j(i,j+1)=2: PRINT AT i+
4,j+4;"█": PRINT AT i+5,j+3;"█":
 LET j(i+1,j)=2
1125 GO TO 1145
1130 IF poz<>3 THEN GO TO 1140
1135 LET j(i+1,j)=2: PRINT AT i+
5,j+3;"█": PRINT AT i+4,j+2;"█":
 LET j(i,j-1)=2
1138 GO TO 1145
1140 LET j(i,j-1)=2: PRINT AT i+
4,j+2;"█": PRINT AT i+3,j+3;"█":
 LET j(i-1,j)=2
1145 FOR k=1 TO 2
1150 INPUT (k);". kettes hajo he
lyzete: ";poz: INPUT "Sora: ";i,
"oszlopa: ";j
1155 LET tip=2: GO SUB 5040
1165 IF hiba=1 THEN GO TO 1205
1170 LET j(i,j)=2: PRINT AT i+4,
j+3;"█": IF poz<>1 THEN GO TO 11
85
1175 LET j(i-1,j)=2: PRINT AT i+
3,j+3;"█": GO TO 1210
1185 LET j(i,j+1)=2: PRINT AT i+
4,j+4;"█": GO TO 1210
1205 LET k=k-1
1210 NEXT k
1215 FOR k=1 TO 3: INPUT (k);".
egyes hajo sora: ";i;" oszlopa:
";j
1220 LET tip=1: GO SUB 5040
1230 IF hiba=1 THEN GO TO 1240
1235 LET j(i,j)=2: PRINT AT i+4,
j+3;"█": GO TO 1245
1240 LET k=k-1
1245 NEXT k
1265 IF taj=10 OR tag=10 THEN GO
 TO 6500
1267 GO SUB 1400: INPUT "LOVESED
 sora: ";i;" oszlopa: ";j
1268 GO SUB 8000: LET ag=i+4: LE
T bg=j+17
1275 IF FN a(i,j)>0 THEN GO TO 1
286
1280 IF g(i,j)=1 THEN GO TO 5700
1282 LET l$="·": GO SUB 6100
1284 GO TO 1290
1286 LET x$="hibas beadas": GO S
UB 6200
1288 GO TO 1266
1290 IF taj=10 OR tag=10 THEN GO
 TO 6500
1295 GO SUB 1400: IF kerul=1 THE
N GO TO 1314
1300 LET m=INT (RND*100)
1305 LET a=INT (m/10)+1: LET b=F
N m(m,10)+1
1310 IF j(a,b)=1 THEN GO TO 1300
1314 GO SUB 6110: IF j(a,b)=0 TH
EN PRINT AT a+4,b+3; OVER 1;"?":
 GO TO 1316
1315 PRINT AT a+4,b+3; OVER 1;"*
"
1316 PAUSE 0: LET q$=INKEY$: PRI
NT AT a+4,b+3;"L"
1320 IF q$="i" THEN GO TO 5250
1325 IF q$="n" THEN GO TO 5200
1330 IF q$="e" THEN GO TO 5400
1335 BEEP 5/10,5: GO TO 1314
1400 FOR c=0 TO 3: PRINT AT 17+c
,0;"
       ": NEXT c: RETURN
5000 LET a=INT (m(i)/10)+1: LET
b=FN m(m(i),10)+1: RETURN
5040 LET aa=i: LET bb=j: GO SUB
5100
5046 IF hiba=1 OR tip=1 THEN RET
URN
5050 IF poz>1 THEN GO TO 5054+10
*(poz-2)
5054 LET aa=i-1: LET bb=j: GO SU
B 5100
5060 IF hiba=1 OR tip=2 THEN RET
URN
5064 LET aa=i: LET bb=j+1: GO SU
B 5100
5070 IF hiba=1 OR tip=2 THEN RET
URN
5072 IF poz=1 THEN RETURN
5074 LET aa=i+1: LET bb=j: GO SU
B 5100
5080 IF hiba=1 OR tip=2 OR poz<3
 THEN RETURN
5084 LET aa=i: LET bb=j-1: GO SU
B 5100
5090 IF hiba=1 OR tip=2 OR poz<4
 THEN RETURN
5094 LET tip=2: GO TO 5054
5100 LET hiba=1: IF FN a(aa,bb)>
0 THEN RETURN
5105 FOR c=-1 TO 1: FOR d=-1 TO
1
5115 LET e1=aa+c: LET e2=bb+d
5120 IF FN a(e1,e2)>0 THEN GO TO
 5130
5125 IF j(e1,e2)>0 THEN RETURN
5130 NEXT d: NEXT c: LET hiba=0
5140 RETURN
5200 GO SUB 1400: IF j(a,b)=2 TH
EN GO TO 7000
5210 LET j(a,b)=1: LET l$="·": L
ET ag=a+4: LET bg=b+3: GO SUB 61
00: IF kerul=1 THEN GO SUB 5500:
 LET a=ak: LET b=bk
5215 GO TO 1265
5250 IF j(a,b)<>2 THEN GO TO 700
0
5255 LET elso=elso+1: LET kerul=
1: LET tag=tag+1: GO SUB 6450: I
F kor=0 THEN GO TO 7000
5260 LET l$="█": LET ag=a+4: LET
 bg=b+3: GO SUB 6100: LET j(a,b)
=1
5265 LET a1=a: LET b1=b
5270 IF elso=2 THEN GO TO 5280
5275 LET aki1=a: LET bki1=b: GO
TO 5285
5280 LET aki2=a: LET bki2=b
5285 GO SUB 5500: LET a=ak: LET
b=bk: GO TO 1265
5400 IF j(a,b)<>2 THEN GO TO 700
0
5402 LET tag=tag+1: LET kerul=0:
 GO SUB 6450: IF kor=1 THEN GO T
O 7000
5405 LET l$="█": LET ag=a+4: LET
 bg=b+3: GO SUB 6100
5410 LET j(a,b)=1: LET aki=a: LE
T bki=b: GO SUB 6400
5415 IF elso>0 THEN LET aki=aki1
: LET bki=bki1: GO SUB 6400
5420 IF elso>1 THEN LET aki=aki2
: LET bki=bki2: GO SUB 6400
5425 LET elso=0: GO TO 1265
5500 IF elso=2 THEN IF aki2=aki1
 THEN GO TO 5540
5501 FOR c=0 TO 1
5510 LET ak=a1: LET bk=b1-1+2*c
5515 IF bk<1 OR bk>10 THEN GO TO
 5535
5530 IF j(ak,bk)<>1 THEN RETURN
5535 NEXT c: IF elso=2 THEN IF b
ki1=bki2 THEN RETURN
5540 FOR c=0 TO 1
5545 LET ak=a1-1+2*c: LET bk=b1
5550 IF ak<1 OR ak>10 THEN GO TO
 5560
5555 IF j(ak,bk)<>1 THEN RETURN
5560 NEXT c: LET a1=aki1: LET b1
=bki1: GO TO 5500
5700 FOR c=-1 TO 1: FOR d=-1 TO
1: LET at=i+c: LET bt=j+d: IF c=
0 AND d=0 OR FN a(at,bt)>0 THEN
GO TO 5730
5725 IF g(at,bt)=1 THEN GO TO 59
00
5730 NEXT d: NEXT c: GO TO 6000
5900 LET x$=" TALALT ": GO SUB 6
200
5905 LET l$="▒": GO SUB 6100
5910 LET g(i,j)=0: LET taj=taj+1
: GO TO 1290
6000 GO SUB 1400: PRINT AT 18,10
;" ELSULLYEDT ": FOR f=1 TO 10:
BEEP 1/100,1: NEXT f: PAUSE 200
6005 LET l$="▒": GO SUB 6100
6010 LET g(i,j)=0: LET taj=taj+1
: LET talalt=0: GO TO 1290
6100 PRINT AT ag,bg;"L";CHR$ 8;
OVER 1;l$: RETURN
6110 PRINT AT 18,2;"TALALT ?"
 nem/n/ igen/i/  elsullyedt/e/"
: RETURN
6200 GO SUB 1400: PRINT AT 18,10
;x$: PAUSE 200: RETURN
6400 FOR c=-1 TO 1: FOR d=-1 TO
1: LET sulya=aki+c: LET sulyb=bk
i+d: IF FN a(sulya,sulyb)>0 THEN
 GO TO 6415
6410 LET j(sulya,sulyb)=1
6415 NEXT d: NEXT c: RETURN
6450 LET kor=0: FOR c=-1 TO 1: F
OR d=-1 TO 1: IF c=0 AND d=0 OR
FN a(a+c,b+d)>0 THEN GO TO 6460
6455 IF j(a+c,b+d)=2 THEN LET ko
r=1: RETURN
6460 NEXT d: NEXT c: RETURN
6500 CLS : IF tag=10 THEN GO TO
6510
6505 PRINT AT 18,10;" NYERTEL ":
 GO TO 6515
6510 PRINT AT 18,10;" NYERTEM "
6515 PAUSE 200: GO SUB 1040: PRI
NT AT 20,10;"JATSZOL MEG (i)": PAU
SE 0: LET a$=INKEY$: IF a$="i" T
HEN GO TO 30
6516 STOP
7000 CLS : PRINT AT 10,10;" CSAL
TAL ! ": GO TO 6515
8000 FOR t=0 TO 30: BEEP 1/100,(
30-t/5): NEXT t: FOR t=1 TO 3: B
EEP 1/100,5: NEXT t: RETURN
```

A program fő részei a következők:

–40	user grafika + alapértékek beállítása
50–130	a hat hajó elhelyezkedése
140–160	a bázismezők lefoglalása
170–1020	a hajók „felépítése"
1040–1046	a gép teljes mezejének kirajzolása
1047–1071	a játékmezők megjelenítése a képernyőn
1080–1245	a játékos hajóinak elhelyezése
1265–1288	a játékos „lő"
1290–1315	a gép „lő"
1316–1335	a játékos válasza
5000	sor- és oszlopmeghatározás
5040–5140	a játékos hajóinak elhelyezését ellenőrzi
5200–5215	a gép nem talált
5250–5285	a gép talált
5400–5425	a gép süllyesztett
5500–5560	találat utáni körbelövés
5700–6010	a játékos lövésének kiértékelése
6400–6415	az elsüllyedt hajó környezetének letiltása
6450–6460	tévedés, ill. csalás ellenőrzés

BOGOS ZOLTÁN

SZÁMSZÖV

COMPUT-80®

mikroszámítógép-családra készített rendszereink:

– főkönyvi és folyószámla-könyvelés
– díjbeszedő vállalatok díjbeszedési és folyószámláinak vezetése
– kereskedelmi rendszer
– pénztárgépes hálózat

COMPUT-80®

központi számítógéppel

– tanácsigazgatási alkalmazási szoftverek
– munkaügyi rendszer
– bérrendszer
– készletnyilvántartási rendszer

ÁLTALÁNOS SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÁSAINK:

– adatrögzítés
– rendszerek üzemeltetése
– rendszerszervezés
– programozás
– mikroszámítógépes rendszerek fejlesztése
– többfelhasználós mikroszámítógépes rendszerek (hardver + szoftver) bérbeadása

További felvilágosítással szívesen állunk rendelkezésükre a 131-072-es telefonszámon, illetve levélben.

SZÁMSZÖV
SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET
1430 Budapest,
Pf. 16.

A SW.'86 kiállításon személyesen is találkozhatunk!

Döntő kérdés:

Mi van a háttérben?

Mert a Commodore számítógépekkel valóban minden feladat megoldható, csak megfelelő kapacitású adattárolóra és jó szoftverekre van szükség.

Mi ezt a megfelelő hátteret kínáljuk!

Rövid határidőre, garanciával vállaljuk

– nagy teljesítményű háttértárak (6 Mb, 10 Mb, 20 Mb) hardver–szoftver illesztését,

– nagygépes kapcsolat megvalósítását (gyors adatátvitel, hardver–szoftver illesztés),

– paralel printer illesztését mikrogépekhez

– adatrögzítési munkák célszoftverjeinek készítését,

– csoportos adatrögzítő rendszerek készítését (2–8 db C 64, 10–25 Mbájt)

– termelésirányítási, irányítási feladatok programozását,

– a gazdálkodást és az ügyvitelt támogató programok készítését.

SW-'86

PRÓBÁLJA KI szoftvertermékeinket, működő rendszereinket!

Bemutatótermünk címe: VERTIKUM-SZÁMÍTÁSTECHNIKA
Budapest 1116. Fehérvári út 106/b.
Telefon: 45-02-45
Telex: 22 4349 VERTI-h

Kívánságára fejlesztéseinkről rendszeresen küldünk tájékoztatást.

KEDVES OLVASÓINK!

Igéretemnek megfelelően a rovatot megosztottam Votisky Zsuzsával és Simonyi Endrével, akiktől egyenként két-két rovatnak megfelelő mennyiségű levelet kaptam azzal, hogy azokat feltétlenül meg kell jelentetni. Ez azonban ebben a formában nem megy. Valami mások törjük a fejünket. Például azon, hogy olyan rovatot létesítünk, amelyben az olvasók ismertetnek érdekes javaslatokat, hardver- és szoftvermegoldásokat. Ezekhez az írásokhoz ábrákat, esetleg fényképeket is kérünk, és azt különösen, hogy az írások ne csak okosak, de érdekesek is legyenek.

Bakos Ottó, Tompa,

Beloiannisz tér 10. 6422

Plébános vagyok, sok vizsgát tettem már életemben, de az első sikertelen próbálkozásom a TV-BASIC vizsga volt. Csúfos „bukásom" azonban nem vette el a kedvemet a további tanulástól, sőt messzemenő terveim vannak a számítógép egyházi vonalon való alkalmazásának kidolgozására. De előbb a TV-BASIC vizsgaanyagát szeretném megvásárolni. Több város könyvesboltjaiban érdeklődtem utána, de sajnos eredménytelenül. Az volna tehát a kérésem, hogy szíveskedjenek részemre a 4 füzetből 1–1 példányt utánvéttel elküldeni.

Levelét köszönöm, a vizsgaanyagot postán elküldtem. A következő vizsga valószínűleg 1986. március 19-én lesz, remélem, akkor találkozunk.

Nyíri Lajos, Mezőtúr,

Szabadság tér 29. 5400

Lapjukhoz hozzájutni művészet! A Postánál még a mai napig is az a válasz, hogy nem lehet előfizetni. Mezőtúr pedig, amely közel 25 ezer lakosú város, ahol főiskola, két nagy létszámú középiskola, számos nagyüzem működik, kevés példányt kap – ha igaz a hír, mindössze tizet.

Csatlakozva szerény lehetőségeinkkel a számítástechnika terjesztéséhez, tájékoztatásul közlöm, hogy a város Művelődési Központja a Volán Ifjúsági Klubjával közösen alapfokú TV-BASIC tanfolyamot indít februártól, 40 órás időtartammal, csütörtöki napokon, 3–3 órás foglalkozásokkal.

A lapterjesztés joga a Postáé, és közismert, hogy egy Posta van. Egész biztos, hogy a lapra elő lehet fizetni, már 1984-ben is volt előfizetés. Én mindenkit arra biztatok, hogy fizessen elő, különben nem tudja megszerezni az újonnan megjelent számokat, amelyek sajnos változó időpontokban kerülnek az újságárusokhoz (és ebben legkevésbé a szerkesztőség a vétkes). Minden új µklubnak örülünk, mert tervezzük, hogy mind a µM előfizetőinek, mind pedig a µkluboknak 1986-ban különféle kedvezményeket fogunk adni.

Lancz István, 92532 Velka Maca Leninova 639,

Csehszlovákia

Engedje meg, hogy így levélben gratuláljak Önnek és az NJSZT-nek ahhoz, hogy a TV-BASIC tanfolyamot megrendezték. A tanfolyam szerintem minden érdeklődőnek javára vált; a magam részéről nem vagyok éppen kezdő a BASIC-ben, mégis sokat tanultam belőle. Remélem, hogy a vizsgák sikeresek lesznek, és hogy később esetleg más, fejlettebb programnyelvekből is láthatunk tanfolyamot a tévében.

Nem tudtam megállni, hogy egy külföldről érkezett elismerést ne idézzek a rovatban. A többi kérdésre levélben válaszoltam.

Abaffy Attila, Pécs,

József Attila u. 32/1. 7. em. 28. 7623

Engedjenek meg két kritikai megjegyzést: 1. Szinte minden példányban található nyomási hiba, amely elsősorban a printerrel kiírt programlistáknál jelentkezik. Az 1985/2. számban például a 6. oldalon szinte olvashatatlan a lista. 2. Nem hiszem, hogy 30 forintos ár mellett a lapnak szüksége lenne arra, hogy értékes cikkek helyett reklámok kerüljenek bele. Ez az arány most 80 : 20 százalék a cikkek javára. Én a reklámot túl soknak tartom, bár elismerem, hogy még mindig kevesebb, mint a nyugati folyóiratokban. De miért kellene nekünk a rossz gyakorlatot tőlük átvenni?

1. Ha nem nyomtatott programlistákat közölnénk, még több lenne a hiba, egyszer már kipróbáltuk. 2. Téved. Ha nem lenne reklám, akkor kétszer annyiért kellene a lapot adnunk. A reklám jó gyakorlat, az olvasók érdekét szolgálja.

Romhányi Mihály, Szolnok,

Ady Endre u. 16. 5000

Már tíz éve foglalkozom számítástechnikával, egy TPA nagygépen dolgozom. Ennek ellenére sok hasznos, új ötletet olvashattam lapjukban, és ami külön öröm számomra, ezek nagy részét munkám során már sikeresen alkalmaztam is. Számomra az újdonságot a számítógépes grafika jelenti, mivel a TPA-n erre nincs lehetőség. Szeretném, ha több grafikai megoldást közölnének, bővebb elemzéssel.

Mi is szeretjük a grafikát, egyre több ilyen cikket, feladatot szeretnénk közölni.

Fodor Dénes, Putnok,

Bartók Béla u. 1. 3630

7. osztályos, 13 éves vagyok. Másfél éve foglalkozom a számítástechnikával. Van egy 48 k-s ZX-Spectrumom.

Egy szokatlan kéréssel fordulok Önhöz: szeretnék az NJSZT tagja lenni. Kérem, írja meg, milyen kötelezettségekkel jár a tagság, milyen lehetőségei vannak egy tagnak stb. Ezenkívül tudnék cikkeket is írni különböző témákról. A µM-nal kapcsolatos észrevételeim: 1. Az 1985/2. szám 7. oldalán levő programot beírtam, de hibás. A hibát csak READ – DATA sorokkal tudtam kijavítani. 2. Nagyon tetszik a Sakkprogramozás rovat Lépésről lépésre című cikke.

Egyre többen szeretnének a Neumann Társaság tagjai lenni, aminek tulajdonképpen nincs akadálya. Forduljanak az NJSZT Titkárságához (1368 Bp., Pf. 240), ahonnan belépési nyilatkozatot és csekket kapnak. A teljes tagsági díj 180 forint, a pártoló tagság díja 50, a diáktagságé 20 forint. A teljes jogú tagok a tagsági díj befizetése után automatikusan kapják a µM-t vagy a Számítástechnikát, az évkönyvet (kétévenként) és a rendezvénynaptárt. A többiek csak az utóbbit. Várjuk a jelentkezéseket.

Ing. Tóth László,

92901 Dunajska Streda Sportova, 1532/2

Csehszlovákiából fordulok Önökhöz egy nagy kéréssel. Iskolánk egy SMEP 50–50 típusú számítógéppel rendelkezik, amely ez év szeptemberétől fog üzemelni. Ehhez el kell sajátítani a BASIC programozási nyelvet. Lelkes nézője vagyok a tévéadásoknak, amelyek az említett nyelv elsajátításához szükséges alapokat ismertetik. Sajnos képernyőről és csak ilyen úton az ismeretek birtokába kerülni nagyon nehéz, talán nem is lehet, még a legnagyobb jóakarat és igyekezet mellett sem. A TV-BASIC tankönyvet ugyan megrendeltem a Magyar Kultúránál, de sajnos ez ideig még nem sikerült megszereznem.

A könyvet elküldtem, a levelet köszönöm.

Nagy István Csaba, Paks,

Árnyas u. 2. 7030

A mai napon kaptam kézhez a TV-BASIC vizsgám sikertelenségéről beszámoló levelüket. Ennek apropójaként engedjenek meg néhány megjegyzést.

Úgy érzem, sikerült a vizsgázók egy részét kissé félrevezetniük, illetve biztosítani a nem egyenlő esélyekkel indulás feltételeit. Ha ez az akció a tömegek szélesebb körű bevonására épült, miért kellett szerepeltetni a feladatlapon olyan kérdéseket, amelyek gépfüggőek? Akinek nincs gépe, annak is választania kell a vizsgán egy géptípust, és csak totózni tud az ilyen típusú kérdésekre. A 16. tévéadásban is elhangzott, hogy a könyv anyagából kell vizsgázni, de a feladatlapon több olyan kérdés is volt, amelynek programjaiban a könyvnek kizárólag a kulcsszótartalomjegyzékében szereplő függvények halmozottan is szerepeltek, holott leírásuk csak egyéb szakkönyvekben található meg! Nem érzem sportszerűnek az ÖTLET-ben kiadott mintafeladatok összeállítását sem, mert az korántsem tükrözte – sőt alábecsülte – a vizsga tényleges nehézségi fokát. És mindezek mellett kellett teljesíteni minimum 80 százalékot, azaz 60 helyes választ!

Én mindezek ellenére úgy érzem, hogy nem szerepeltem rosszul, de a pontszámomat sajnos az Önök tájékoztatójából nem tudtam meg!

Meglepetéssel tapasztaltam, hogy vizsgán tett igéretükkel ellentétben a feladatlapomat nem kaptam meg, amely – akárhogyan is sikerült – a későbbiekre nézve hasznosan segítené felkészülésemet.

Levelére részben az ÖTLET július 18-i számában válaszoltam. A programozói vizsga a gyakorlat vizsgája, ezért szervezték a µklubokat. Gyakorlat nélkül – ezt korábban is elmondtuk – általában nem tudtak a jelöltek levizsgázni. A feladatlapot nem adjuk vissza, miután az a vizsga dokumentuma. Minden mikroklubnak küldünk feladatlapot, hogy gyakorolni tudjanak.

Török Péter, Budapest,

Dési Huber u. 15. 1098

Megítélésem szerint kevés a lapban az ismertetett hardver áramkörök száma. Nagy hátrány, hogy (eltekintve a LOGO nyelv igen rövid ismertetésétől) csak BASIC nyelvű programokat közölnek, holott szeretném szükség lenne FORTRAN, Pascal, ALGOL nyelvű programok közlésére is. Ez igen fontos volna, mivel a műszaki egyetemen főleg Pascal és FORTRAN nyelveket oktatnak.

A µKlub rovat gyakorlatilag hardvercikkeket közöl. Egyébként a lap tükrözi a hardver- és szoftveramatőrök arányát. A lap a mikroszámítógépekkel foglalkozik, és nem a nagyszámítógépekkel, így nem közölhetünk Pascal vagy LOGO programokat. Az előbbi csak a professzionális személyi számítógépeken fut, az utóbbi egyik hazai (HT vagy Primo) gépen sem. Így marad a BASIC. Én azon is elgondolkoznék, hogy alternatív módon nem lehetne-e a műegyetemen BASIC-et is oktatni, akkor csak nagyobb lenne a gyakorlási lehetőség?!

Surányi Tamás, Orosháza,

Kond u. 57/c. 5900

Valóban megkezdik-e a számítógépprogramok sugárzását a rádióban (úgy tudom, hogy egyelőre csak Spectrum programokét), és ha igen, akkor mikortól várható ez? Lesz-e még valamilyen számítógépes tévéműsor?

Valóban tárgyalunk erről a Rádió illetékes vezetőivel. Egy a tévéműsorhoz hasonló rádiós oktató- és magazinműsort javasoltunk, amelyhez a programokat és az oktatási anyagot is így sugároznák, a minőségi

követelmények miatt csak az URH sávon. Hogy miért? Ha rajtunk múlna, akkor holnap. Először valóban Spectrum programokkal kezdenénk.

Az új műsor címe: Bevezetés a számítástechnika alkalmazásába. Ezen a műsoron már dolgozunk. Kevésbé előrehaladott állapotban van a Szoftvertechnológia című adássorozat. A TV-BASIC-et – remélem – sikerül január 1-től megismételni.

Babilon Nyelvstúdió, Budapest I.,

Várfok u. 2/b. 1012

A Babilon Nyelvstúdió, ahol idegen nyelvek, programozási nyelvek oktatásával, valamint idegen nyelvű műszaki szövegek fordításával foglalkozunk, ez alkalommal másodszor ajánlja fel szolgáltatását a személyiszámítógép-tulajdonosok táborának. (Először – és tudomásunk szerint elsőként – magánérdeklődők számára minimal BASIC tanfolyamot indított.)

Bárki, aki a nyelvstúdiónál akár postán is megrendeli valamelyik kezelési utasítás magyar nyelvre fordítását, az vállalja ugyan, hogy a Közlönyben jóváhagyott fordítási és gépelési díjat kifizeti, ha azonban a kezelési könyvre újabb megrendelés érkezik, a második megrendelő már csak 55%-os fordítási díjat fizet, és az előző megrendelő 50%-ot visszakap az általa fizetett díjból. További megrendelők esetén a visszafizetés folytatódik, úgy hogy a nyelvstúdió a fordítást csak egyszer fizetteti ki a megrendelőkkel. (A másolásra ez értelemszerűen nem vonatkozik.)

Jelen engedményes szolgáltatásunk kizárólag magánfelhasználók részére és személyi számítógép gépkönyvekre vonatkozik.

Hirdetéseket itt nem közlünk, ezért levelüket közérdekű információnak minősítve adjuk közre.

Domokos Zoltán, Hódmezővásárhely,

Bocskai u. 21/a. 6800

Az ÖTLET-ben és a µM-ban olvastam, hogy elfordult, hogy valaki hibás Commodore-t vásárolt. Én most jutottam el odáig, hogy vegyek egy ilyen gépet, de félek, hogy ha hibásat veszek, akkor fújhatom az egészet. Ezért kérdem, hogy egy számítógépen hogy lehet észrevenni a kisebb-nagyobb gyári hibát? Vagy vásárlásnál át lehet nézni a gép BASIC-jét? Milyen hibák fordulnak elő a legsűrűbben a Commodore 64-nél?

Vásárláskor csak nagyon felületesen lehet kipróbálni a számítógépet. Ezért kell olyan gépet venni, amelyet megbízható cég ad el, ad hozzá garanciát, és megmondja, hogy itthon hol javítják.

Kanics Miklós, Budapest,

Völgy u. 31/a. 1021

Tavaly építettem egy HOMELAB 2 (Aircomp–16) típusú kisszámítógépet, amely azóta is kifogástalanul üzemel. Gondom csak akkor van vele, ha idegen magnóval készített kölcsön kazettáról szeretnék beolvasni, vagy új, számomra még ismeretlen szalagot akarok programtárolásra felhasználni. Ilyenkor a gépem makacsul ellenáll, és csak többszöri próbálkozás után sikerül megetetni vele. A helyes magnóbeállítás érdekében készítettem egy rövid programot, amely a képernyőre kirajzolja a magnóbemeneten tapasztalt szintváltozásokat. Mivel sok Aircomp–16 tulajdonosnak lehetnek hasonló gondjai, ezúton közlöm a jól használható kis programot. Egy kis ügyességgel az ismeretlen számítógéppel készített felvételek is megfejthetők, természetesen a LOADER megírása a felhasználó feladata marad.

A program két fő részből áll. A BASIC program a megfelelő változókat állítja be, letárolja a gépi kódú rutint, információt közöl a program használatáról, megrajzolja az időosztásos skálát, és elindítja a gépi kódú rutint. A gépi kódú program végzi a magnóbemenet vizsgálatát és a jelalak kirajzolását a grafikus terület aljára.

Minden egyes gombnyomásra három ms hosszúságú mintát vesz a bemenetről, és azonnal megjeleníti a képernyőn. A „H" és az „L" gombok lehetővé teszik a szinkronizált mintavételt. Az „R" gomb megnyomásával a rutin visszaugrik BASIC-be.

```
MINISZKOP program HOMELAB 2
(Aircomp–16) számítógépre
10 GL=25 : CR=0 : PRINT CHR$(12,5)
   : CR=1 : HM
   =PEEK(16443)+256*PEEK(16444)–72
20 POKE HM, 42, 59, 64, 17, 72, 3, 25, 235, 33, 0,
   224, 223, 254, 82, 200, 50, 62, 62, 254, 72, 32, 10,
   203, 70, 32, 252, 203, 70, 32, 14, 24, 250, 254, 76,
   32, 8, 203, 70, 40, 252, 203, 70, 32, 252, 213, 221,
   225, 6, 40, 62, 1, 203, 14, 23, 48, 251, 221, 119,
   136, 47, 221, 119, 120, 221, 35, 16, 238, 50, 63,
   63, 24, 195
30 POKE 16404, 193, 195 : PRINT ,
   ,"MINISZKOP↓"
40 PRINT " ALKALMAS A MAGNOJELEK
   IDOBENI VIZSGALATARA.↓"
50 PRINT "H=FELFUTO ELRE VEZERELT"
60 PRINT "L=LEFUTO ELRE VEZERELT"
70 PRINT "R=VISSZATERES BASIC-BE"
80 PRINT "BARMELYIK GOMBBAL EGY
   SOR BEVEHETO"
90 PRINT "EGY OSZTAS 100 MIKROSEC.↓↓↓"
100 FOR I=0 TO 319 : PLOT I,15 : NEXT
110 FOR I=0 TO 310 STEP 10 : FOR J=10 TO 14
   : PLOT I,J : NEXT ,
120 FOR I=0 TO 300 STEP 50 : FOR J=6 TO 9
   : PLOT I,J : NEXT, ,
130 FOR I=0 TO 300 STEP 100: FOR J=0 TO 5
   : PLOT I,J : NEXT ,
140 N PRINT "0," 1"," 2"," 3↓↓↓↓"
   : CALL HM : GL=0
```

Nagyon sok olvasónk hiányolja, hogy nem tartunk gépi kódú tanfolyamot lapunkban. Egyelőre még mindig nem, de a µprogramokban és másutt is találhat hasznos rutinokat a szorgos olvasó.

Reméljük egyébként, hogy lassan megjelennek olyan könyvek, amelyek nemcsak Z80-nal, hanem az egyes gépek speciális gépi kódú programozásával foglalkoznak. Például: Bóc István: ZX81 Basic és Assembler (MK 1985.); Sztrókay Kálmán: Z80 Assembler HT–1080Z számítógépes példákkal (MK 1985.); ZX–Spectrum Basic és gépi kódú programozás (Ipari Informatikai Központ).

Békési Gábor, Budapest,

1174

6. osztályos tanuló vagyok, és szeretnék részt venni a TV-BASIC vizsgán. Nagyon örülnék a ZX81 játékprogram-könyvnek, szeretnék benne minél több Assembler listát találni, mert nemrég kezdtem tanulni a Z80 gépi kódját. Mivel minden elérhető klubtól messze lakom, megpróbálom magam megérteni az Assembler programozást. Eddigi tanulásom eredménye egy olyan program, amely a képernyőt egy bizonyos karakterrel tölti fel.

A legtöbb problémát a Newline kivédése okozta. Most a képernyő jobbra, balra és lefelé „scroll"-oztatásával kísérletezem. Ha volna egy ilyen programjuk, kérem küldjék el.

A „feltöltő" program listája a következő:

```
        Assembler      Dec.
        LD HL,(16396)   42
                        12
                        64
        INC HL          35
        LD B,22          6
LOOP 2  LD C,32         14
                        32
LOOP 1  LD (HL),128     54
                       128
        INC HL          35
        DEC C           13
        JRNZ LOOP 1     32
                       250
        DEC B            5
        INC HL          35
        JRNZ LOOP 2     32
                       244
        RET            201
```

Szieberth András, Székesfehérvár,

Sebes I. u. 17/b. 8000

Az Ipari Informatikai Központ ZX-SPECTRUM BASIC és gépi kódú programozás című könyvében találtam egy apró, de bosszantó hibát. Talán többen észrevették, hogy a 8–10. oldalon található gépi rutin meghívásakor „elszáll" a program. Ez természetes, mivel az első (itt 7D00) címre 3E (LD A, N) helyett 06-ot (LD B, N) kell írni.

A fenti rutin a CL-SCROLL. Nyilvánvaló, hogy nem az A, hanem a B regisztere kell tölteni a felfelé léptetendő sorok számát, ahogy ezt helyesen leírják az előző oldalon.

Olvastam, hogy SHARP PC–1245-ös jelzésű „zsebszámítógéphez" leírást keres egyik olvasójuk.

Még 1984 nyarán fizikatanárom hozott Nyugatról egy ilyen gépet. Nem tudta használni, ezért februárban megkért, hogy fordítsam le német nyelvű gépkönyvét, és segítségül kölcsönadta gépét is. Egyből nekiláttam, de nem találtam kielégítőnek a könyv felépítését: olvasóját abc sorrendben szándékozik bevezetni a programozás rejtelmeibe. Ezért kicsit átírtam, szerintem jobb programokat írtam hozzá, mint az eredetiek. Szándékom az volt, hogy a könyv az olvasóját megtanítsa BASIC-ben programozni – persze ha van ilyen „zsebgépe" –, ezért egyes részleteket éppen csak érintek.

Nem tudjuk, hogy a könyv iránt mekkora az igény. A kézirat első részét a szerkesztőség már megkapta.

Molnár László, Győr,

Rákos F. u. 13. 9024

A µM 1985. évi számaiban olvastam arról, hogy klublemezeket kívánnak forgalomba hozni. Szeretnék én is hozzájáruni egy C64-hez készített lemezhez. Kérem, szíveskedjék tájékoztatni arról, hogy a meglévő programok mire szolgálnak, hogy lehet a lemezhez, illetve tartozékokhoz hozzájutni, és mennyibe kerül a kiegészítő elemekkel együtt. Lehetséges-e a lemezhez két botkormányt kapni? Mennyibe kerül a tárbővítés 256 k-ig?

Sok olvasónk érdeklődésére közöljük, hogy a klublemezek a 14. sz. ÜMK-tól (Bp. VII., Rákóczi út 8/a. ügyintéző: Ollé Tibor) megrendelhetők. Áruk botkormánnyal együtt 5000 forint. Ha a lemezt a megrendelő kívánsága szerint kell összeválogatni, az 50 százalék felárat jelent. Külön botkormányt csak klubtagok vásárolhatnak, 1000 forintos áron.

Nádasdy Zoltán, Szigetszentmiklós,

Munkásőr u. 12/d. 2310

A HCC Commodore szekció összejöveteleinek időpontját kéri. Az alábbiakban felsoroljuk a HCC állandó programjainak helyét és időpontját:

– HCC Commodore szekció: BME H épület III. emelet, számítógépszoba. Páros héten csütörtökön 17 órától. BME F épület II. szárny, II. emelet, dr. Kiss László szobája. Páratlan héten kedden 17 órától.

– HOMELAB – Aircomp szekció: Eötvös Klub, Bp. V., Károlyi Mihály u. 6. Minden kedden 17 órától.

– Sinclair szekció: Általános Iskola, Bp. X., Üllői út 118. Minden hétfőn 18 órától. Volán pinceklub, Bp. XI., Ménesi út 2. Minden szerdán 18 órától.

•

Nagyon sok levelet kapunk, hogy küldjünk különböző katalógusokat, játékprogramokat, adjunk meg címeket, főleg külföldi számítógépgyárak címeit, hogy különböző dolgokat tudjanak kérni. Ne haragudjanak a szerkesztőségre, de erre nincs lehetőségünk. Arra sem, hogy bármit kijárjunk vagy elintézzünk. Szeretnénk a „kaptafánál" maradni – ez az Olvasók és a szerkesztőség közös érdeke.

KOVÁCS GYÖZŐ–VOTISKY ZSUZSA
–DR. SIMONYI ENDRE

FÓRUM

Lapunkat kibővítjük egy új szolgáltatással: olyan olvasóink számítógép-ismertetéseit közöljük, akiknek hazánkban ismeretlen gépük van, és ezért magányos felhasználónak érzik magukat. Reméljük, hogy segítségünkkel a „különleges" gépek tulajdonosai egymásra találnak, mások pedig szívesen olvassák a felhasználók véleményét ezekről a számítógépekről. (Az alábbiakban elsőként bemutatott LASER–210-re Játékprogramok rovatunkban közlünk programot.)

LASER–210

A Magyarországon ritka, kevéssé ismert LASER–210 igen könnyen programozható és egyszerűen kezelhető számítógép, amelyet talán leginkább a Spectrumhoz és a HT–1080Z-hez lehetne hasonlítani. Kis méreteivel – 48 × 170 × 285 mm – még egy aktatáskában is elfér a hozzá tartozó magnetofonnal, kazettákkal és hálózati csatlakozókkal együtt. A 8 k-s gép normál 9 V-os adapterrel üzemeltethető, működését piros LED jelzi. A hirtelen kibekapcsolás nem okoz olyan gondokat, mint a HT-n. A televízió- és monitorcsatlakoztatás egyazon kábelen történik, de két csatlakozóaljzaton. A tévé használatánál a gép alján levő kapcsolóval választhatunk az NTSC, illetve a PAL rendszer között.

A perifériák

A géphez a LASER DR–10 kazettás adattároló tartozik, de bármelyik ilyen jackdugóval rendelkező magnót is hozzákapcsolhatjuk. A program kimentésére a CSAVE „név", beolvasására pedig a CRUN vagy a CLOAD utasításokat használhatjuk. Adatok tárolása a PRINT # és az INPUT # utasítással történik. A programtárolás egyszerű és megbízható, ugyanis a VERIFY utasítással még ellenőrizni is lehet.

A géphez 16 és 64 kbájtos memóriabővítő csatlakoztatható a hátsó, közvetlen kimeneten.

A botkormány csatlakoztatása a vele egybeépített interfészen keresztül, a PERIPHERIE kimeneten történik. Csatlakoztatáskor ügyelni kell arra, hogy a gép kikapcsolt állapotban legyen. Az A = INP (43) AND 31 esetén A értéke 31-ig változik a botkormány különböző irányokba állításával. Ha a 43-at 46-ra változtatjuk, akkor a másik kormányról lehet adatot bevinni.

Ugyanerre az aljzatra kapcsolható a plotterprinter is, amelyhez a szokásos utasítások használhatók (LPRINT, LLIST, COPY).

A processzor

A gép NEC 8303X5 típusú, japán gyártmányú processzora Z80-as kódban programozható. Tára 16 kbájt ROM-ot és 8 kbájt RAM-ot tartalmaz a többi tárterületen kívül. Annak ellenére, hogy a gépen „Made in Hong Kong" felirat szerepel, japán alkatrészekből áll.

A billentyűzet

Az enyhén döntött billentyűzet hasonlít a ZX-Spectrumához, de kezelése egyszerűbb. A gombokon látható jeleket, grafikus karaktereket a SHIFT gomb lenyomásával, a felettük levő BASIC utasításokat a CTRL, az alattuk levőket vagy a függvényeket a CTRL-RETURN benyomásával hozhatjuk elő. A gombok lenyomásakor a gép csipogó hangot ad, akárcsak a Primo.

BASIC

A LASER–210 BASIC-je néhány apró kivételtől eltekintve megegyezik a normál BASIC-kel. Az alaputasításokon kívül néhány speciális utasítása is van, például: COLOR, MODE, SOUND.

A programok javítását nem editálással, hanem a kurzor megfelelő sorba és pozícióba állításával lehet elvégezni. A sort előretolni az INSERT, beljebb hozni pedig a ROBOUT gomb lenyomásával lehet.

A képernyő kezelése

A képernyő 32 × 16, azaz 512 karakteres. A kurzor egy normál és inverz alakban villogó SPACE. A képernyőn négy alapszín és négy betűszín alkalmazható (piros, sötét- és világoszöld, narancs). A grafikus jeleket nyolc színben állíthatjuk elő a COLOR utasítással. A karakterek négy részre bonthatók, és használhatók inverz betűk is. A képernyő meghatározott pozíciójába írásra a PRINT @ szolgál, alkalmazható továbbá a PRINT TAB és a PRINT USING is. Sajnos a gép csak nagybetűket ismer.

A grafika

Ha a gépet grafikus üzemmódban akarjuk használni, akkor át kell váltani a MODE (1) utasítással. Így a képernyőt 128 × 64 képpontra bonthatjuk, amelyeket szintén nyolc színben jelenithetünk meg. A képernyőről való beolvasáskor az A = POINT (x, y) utasítást kell alkalmazni. Ilyenkor A értéke az X, Y koordinátákon levő pont színének kódszáma lesz. Rajzolni a SET és a RESET utasítással lehet. Az origó a bal felső sarokban helyezkedik el. A program befejezése után a gép automatikusan normál üzemmódra vált vissza.

A szoftver

Felhasználói szoftvereket általában kazettán lehet beszerezni, ami itthon nem egyszerű. Tudomásom szerint külföldön is csak a Neckermann foglalkozik ezek árusításával. Egyébként a VZ 200-as programjai is futtathatók a gépen.

A programfutás sebességét tekintve a LASER–210 az ismert tesztek szerint a HT–1080Z és az Atari típusokat is megelőzi.

A gép nagy előnye, hogy jóformán egyáltalán nem melegszik fel.

A LASER–210 ára 1984-ben 300 DM körül volt, és a perifériák is viszonylag olcsók.

BÁRTFAI BARNABÁS

TUDÁSPRÓBA

1. Melyik esetben kapunk nagyobb számot, ha a 16-os alapú számábrázolási rendszerben E–C értékét, vagy ha a 10-es alapú számábrázolási rendszerben 14–12 értékét vesszük?

2. Szerepelhet-e egy számológép hardverelemeinek felsorolásában egy mesterséges égitest?

3. Van-e olyan szám, amelynek a 10-es alapú számábrázolási rendszerben két különböző alakja (felírása, kifejtése, előállítása) is van, és a két kifejtésben egyetlenegy közös számjegy sem fordul elő?

4. Van-e olyan szám, amelynek a 10-es alapú számábrázolási rendszerben 3 különböző kifejtése is van?

5. Igaz-e, hogy a 10-es alapú számábrázolási rendszerben a véges tizedestörteknek és azoknak a végtelen tizedestörteknek, amelyeknek jegyei egy jegytől kezdve mind kilencesek, pontosan két különböző kifejtése van, a többi számnak pedig pontosan egy?

6. Lehet-e, hogy egy számnak a 2-es alapú számábrázolási rendszerben két különböző felírása van?

7. Igaz-e, hogy végtelen sok olyan szám van, amelynek a 3-as alapú számábrázolási rendszerben két különböző kifejtése van, úgy, hogy ezekben egyetlenegy a másikban is szereplő számjegy sincs?

8. Mi közе a számítástechnikához az egérnek, a teknősbékának, az almának és a sárgabaracknak?

9. Csak kétfajta műveleti jel, a + és a • jel használatát megengedve, összesen 10 darab műveleti jellel, a V1, V2, ..., V10 változókból legfeljebb hány tagú szorzat, illetve legfeljebb hány tényezős összeg állítható elő a BASIC nyelv szabályait betartva?

10. Mi a preprocesszor?

11. Lehet-e, hogy egy végtelen sorozat tagjai rendre kisebbek, mint egy másik végtelen sor megfelelő elemei?

12. Valószínűségelméleti oktatóprogramokban (is) lépten-nyomon építenek a következő hipotézisekre.

Feldobott (homogén) pénzérme fejre esésének és írásra esésének valószínűsége egyaránt 1/2.

Feldobott szabályos és homogén anyagú kocka adott lapjára esésének valószínűsége 1/6.

A gyakorlatban mind ez ideig senki sem volt képes e feltételezések igazolására, sőt a tapasztalat homlokegyenest e hipotézisek ellen szól. Végezzünk néhány érmefeldobást és kockadobást! Mit tapasztalunk?

13. Helyes-e az eljárások egy csoportjának „aritmetikai és matematikai eljárások" osztályokba sorolása?

14. Helyes-e a következő definíció?

„Az egészrész-függvény definíciója szerint negatív számra is a nála kisebb – abszolút értékben nagyobb vagy egyenlő! – egész szám adódik."

15. Van-e „természetes alapú hatvány"?

16. Mi a hiba a következő definícióban?

„Az e^x függvény értékét kiszámító eljárás egy (kitüntetett) regiszterbe elhelyezett egyszeres pontosságú, lebegőpontos számot a természetes logaritmus alapszámára mint alapra emeli."

17. Igaz-e, hogy a HT–1080Z gép Arcustangens-kiszámító eljárása egy (kitüntetett) regiszterbe lebegőpontos formában elhelyezett „tangens érték inverzét számítja ki"?

18. Igaz-e, hogy „szintaktikus programozási hiba például a rossz helyre írt számjegy és a helytelen műveleti utasítás"?

19. Helyes-e a következő definíció?

„Forrásprogramnak nevezzük a hexadecimális számjegyekkel felírt programot."

20. Gyakran szerepel a számítástechnikai irodalomban a „szimbolikus nyelv" kifejezés. Létezhet-e olyan nyelv, amely nem szimbolikus, azaz nem szimbólumokat használ?

Összeállította TAKÁCSY ILDIKÓ

Varga László: Személyi számítógépek kezelése, programozása és alkalmazása

(Budapest, 1985. Akadémiai Kiadó, 400 oldal. Ára: kb. 180,— Ft.)

Ez a könyv azoknak nyújt segítséget, akik már elsajátították egy személyiszámítógép-típus kezelését, és egy másik típusú gépet is akarnak használni. A hazánkban ismert eltérő gépek ugyanis általában a BASIC nyelv valamely – a többitől különböző – „nyelvjárását" ismerik. Az átállás nem könnyű. Ezt a nem mindennapi feladatot oldja meg a könyv úgy, hogy alternatív programokat mutat be a Commodore 64, az ABC–80, a HT–1080Z, a Texas 99 és a Proper 16 személyi számítógépekre alkalmazható feladatok elvégzésére.

A szerző az alapmegoldást a Commodore 64-re dolgozza ki, a többi típus eltérő programozását ezzel párhuzamosan, a tükröroldalon mutatja be. A programokat és kezelési utasításokat az említett számítógépeken tesztelték, működésük szavatolt.

A könyv nemcsak oktatóprogramot tartalmaz, hanem számos olyan szoftveranyagot is, amelyek a gyakorlatban jól használhatók: statisztikai, pénzügyi, üzemgazdasági, perifériakezelő stb. programokat.

Ehhez hasonló jellegű munka jelenleg nem található a könyvpiacon.

Dr. Rózsa Lajos (szerk.): Mikroszámítógép-alkalmazási esettanulmányok

(Budapest, 1985. SZÁMALK, 400 oldal. Ára: kb. 250,— Ft.)

A legkülönbözőbb rendeltetésű berendezésekben viharos gyorsasággal megjelenő mikroszámítógépek egyik igen fontos alkalmazási területe az ipari termelési folyamatok ellenőrzése és automatikus irányítása. Ezt számos sajátosság (pl. a feladatok rendkívüli sokrétűsége, bonyolultsága, realtime jellege stb.) különbözteti meg az egyéb mikroszámítógép-alkalmazásoktól (pl. személyi számítógépek, ügyviteli alkalmazások stb.). Az ipari termelés irányításában betöltött szerepük következtében jelentős mértékben ezek a berendezések és a velük megvalósított automatizálás színvonala határozzák meg a gyártott termék mennyiségét, minőségét, és a gyártással járó költségeket.

A könyv egy, az MTA Számítástechnikai és Automatizálási Kutató Intézetében ipari irányítási feladatok ellátására kifejlesztett eszköz, a Mikroszámítógépes Folyamatirányító Berendezés (MFB) konkrét alkalmazási példáinak ismertetésével kívánja bemutatni ennek a technikának az alkalmazási lehetőségeit.

Tíz fejezetben ismerteti az MFB felépítését és azt a programrendszert (alapszoftvert), amelynek felhasználásával a gép alkalmassá vált az ipari mérésadatgyűjtési és irányítási feladatok megoldására. Hat önálló fejezete egy-egy konkrét feladat megoldásának részleteivel ismerteti meg az olvasót. További egy-egy fejezete azt mutatja be, hogy a berendezés önálló programfejlesztő eszközként is jól használható, illetve az ipari irányítási feladatok mellett más célokra, pl. sportesemények eredményhirdető berendezéseinek működtetésére és az ezzel kapcsolatos adatnyilvánításra és feldolgozásra is alkalmas.

A szerzőgárdát azok a szakemberek alkotják, akiknek közvetlenül szerepük volt a berendezés (hardver) és az alapszoftver kidolgozásában, illetve az adott irányítási feladat megoldásában.

Ligeti Gábor–Szervánszky György: A ZX Spectrum programozása

(Budapest, 1985. SZÁMALK, 250 oldal. Ára: kb. 80,— Ft.)

A könyv a ZX Spectrumot használók széles táborához szól. A szerzők úgy dolgozták fel a témát, hogy a könyvet mindenki, aki kapcsolatba kerül a ZX Spectrummal, használni tudja. A könyv az ebben a témában megjelent könyvektől eltérő, új megközelítésben vezet végig a tudnivalókon, hiszen a különböző számítástechnikai ismeretek más-más olvasási módot igényelnek.

Az első fejezetek a ZX Spectrum felépítésével, üzembe helyezésével és BASIC nyelvű programozásával foglalkoznak. A számítástechnikával most ismerkedők a gép működőképessé tételétől a kész programok használatán át eljuthatnak a számítógép önálló alkalmazásáig, a programozásig.

A BASIC programozási nyelvet ismerők az egyes utasítások leírásából megállapíthatják, mennyire tér el az általuk ismert BASIC-változat a ZX Spectrum BASIC nyelvétől. A könyv második része a számítástechnikában jártas olvasóknak szól. A haladóknak szánt, gépi kódú programozásról írt fejezetek a ZX Spectrum lelkét jelentő Z80 mikroprocesszor programozásába vezetik be a felhasználót. Ezt a részt a Z80 programozásával foglalkozóknak is ajánljuk, mivel számtalan hasznos ROM rutin címét, működési körülményeit ismerhetik meg belőle.

Végül megjegyezzük, hogy a könyv ugyan a ZX Spectrum programozását írja le, de a BASIC utasításokat leíró részt a ZX 81 felhasználók is eredményesen használhatják.

Kernighan, B. W.–Ritchie, D. M.: A C programozási nyelv

(Budapest, 1985. Műszaki Könyvkiadó, 231 oldal. Ára: 72,— Ft.)

A C általános célú programnyelvet a tömörség, a korszerű vezérlési és adatstruktúrák használata, a bőséges operátorkészlet jellemzi. Nem nevezhető sem nagyon magas szintű, sem nagy nyelvnek, és nem kötődik egyetlen speciális alkalmazási területhez sem. Ugyanakkor a megkötések hiánya, az általános jelleg sok magas szintű nyelvnél kényelmesebbé és hatékonyabbá teszi.

A C nyelv nem kötődik szorosan egyetlen hardverhez vagy rendszerhez sem. Könnyen írhatunk olyan programokat, amelyek változtatás nélkül futnak bármely más, a C nyelvet támogató gépen.

A könyv a C nyelvű programozás elsajátításához ad segítséget. Az olvasó már az Alapismeretek című fejezet megértése után elkezdhet programozni. A könyv ezután külön fejezetekben ismerteti a C nyelv fő összetevőit, majd referencia-kézikönyv formájában is összefoglalja a nyelvet. Ez a rész főleg példaprogramok írásából és módosításából áll, nem száraz szabványgyűjtemény.

A könyv nem bevezető jellegű programozási segédkönyv, feltételezi, hogy az olvasó ismeri a programozás alapfogalmait, de útmutatásai alapján egy kezdő programozó is megtanulhatja a nyelvet, bár szüksége lehet jártasabb kolléga segítségére.

Tapasztalatok bizonyítják, hogy a C sokféle feladat megfogalmazására alkalmas, kellemes, kifejező és rugalmas nyelv.

Dr. Kovács Magda: Egyszerűen a mikroszámítógépekről

(Budapest, 1985. LSI Alkalmazástechnikai Tanácsadó Szolgálat, 322 oldal. Ára: 156,— Ft.)

A könyv áttekintést nyújt a mikroelektronika és a mikroszámítógépek fejlődéséről, a mikroszámítógépek műszaki jellemzőiről. Főbb fejezetei: A mikroelektronika fejlődési tendenciái, műszaki-gazdasági kérdések; Mikroszámítógépek a gyakorlatban; Alapfogalmak; Mikroszámítógép; Programozási alapismeretek; Mikroprocesszor alapú rendszer; Mikroszámítógépek programozási kérdései; Utasításkészlet; A/D és D/A átalakítási módszerek. (A könyv kritikájára a későbbiekben még visszatérünk.)

Legyen a µMagazin munkatársa!

MIKROSZÁMÍTÓGÉP MAGAZIN

Ossza meg tudását az Önhöz hasonló érdeklődésűekkel! Vegyen részt a mikroszámítógépes lavina hazai kibontakoztatásában!

A Mikroszámítógép Magazin munkatársakat keres. A lap várható terjedelembővülésével összefüggésben minden olyan jelentkezőt szívesen látunk, aki a mikroszámítógépekkel kapcsolatos valamelyik területen megfelelő jártassággal rendelkezik. Ilyen témák például:

- népszerű személyi számítógépek gépi kódú programozása
- alkalmazásgenerátorokra és generikus szoftverekre (electronic spreadsheet, word processor stb.) alapozott alkalmazásfejlesztés
- mikroszámítógépek alkalmazása az adat- és információfeldolgozásban
- játékszoftverek készítése
- számítógépes szakismeretek oktatása
- mikroszámítógépes piac
- szoftvertechnológiák és fejlesztő nyelvek
- számítógéppel támogatott oktatás
- operációs rendszerek és használatuk
- mikrogépes perifériák
- helyi hálózatok

és sok egyéb téma, amit még felsorolni is igen nehéz lenne (folyamatirányítás, robotok, műszaki alkalmazások stb.). Jelentkezését várja a µMagazin szerkesztősége.

VÁLASZOK A TUDÁSPRÓBA KÉRDÉSEIRE

1. Egyik esetben sem, mert mindkét különbség értéke 2.

2. Igen, például akkor, ha a gépet felhasználók távközlési műhold segítségével (közvetítésével) adják és kapják az információt.

3. Van, mégpedig végtelen sok. Egy példa az 1 és a 0,999…, amelyeknek értéke azonos, de jegyeik különbözők. Ilyen számpárok egy végtelen sorozata pedig a következő

1 és 0,999…
10 és 9,999…
100 és 99,999…
…

4. Nincs. És ebből következik, hogy olyan szám sincs, aminek kettőnél többféle (például négyféle) egymástól különböző kifejtése volna a 10-es alapú számábrázolási rendszerben. (A bizonyítást végezze el az Olvasó! A tétel nemcsak 10-es alapú számábrázolási rendszerre igaz.)

5. Igaz. (A bizonyítást végezze el az Olvasó! Hasonló tétel nemcsak a 10-es alapú számábrázolási rendszerre igaz.)

6. Igen, például 1,000… és 0,111…, de olyan szám nincs, aminek két különböző előállítása van, és ezekben egyetlen közös jegy sem fordul elő. Ha ugyanis az egyik tartalmaz egy egyest, a másik csak nullát tartalmazhat, tehát értéke is nulla kell, hogy legyen. A nulla előállításában pedig nem szerepelhet egyes.

7. Igaz, például az

1 és 0,222…
10 és 2,222…
100 és 22,222…
…

8. Az egér (mouse, ejtsd *máusz*) információbejuttató eszköz. A teknősbéka (turtle, ejtsd *tötl*) a LOGO nyelvű programozás egyik kelléke. Az alma (apple, ejtsd *epl*, ahol az e olyan, mint a magyar ember szó első e hangja) – pontosabban az Apple – számológépgyártónak is és géptípusoknak is neve. A sárgabarack jelentésű apricot (ejtsd *eiprikat*) egy személyi számológéptípus neve.

9. Semennyi, mert *összegnek tagjai*, *szorzatnak* pedig *tényezői* vannak.

10. A processzor jelentése *eljárást, feldolgozást végző, feldolgozó*. A preprocesszor jelentése processzor előtti, ill. előfeldolgozást végző, előfeldolgozó.

Előfordul, hogy egy feldolgozás több és általában nagyobb, önálló részekre bontható. Ilyenkor, ha több részre bontják a feldolgozást, az először sorra kerülő művelet (műveletek) elvégzését előfeldolgozásnak is hívják. Az előfeldolgozást az előfeldolgozó, az ún. preprocesszor végzi.

Használatos a szó puszta viszonyításnál is, amikor valamilyen M művelet előtti, az M műveletet előkészítő műveleteket végző berendezést, berendezésegyüttest hívnak az M művelet (illetve az ezt elvégző berendezés) preprocesszorának. Tulajdonképpen csak *ez a viszonyító értelmű szóhasználat a helyes*, amikor is *valaminek a preprocesszoráról* (illetve posztprocesszoráról) beszélünk. Eszerint ha A a B (egy) preprocesszora, akkor B az A-nak (egy) posztprocesszora.

Ha az A, a B és a C eljárás ebben a sorrendben követi egymást, akkor a B egyszerre posztprocesszora az A-nak, és preprocesszora a C-nek. Furcsa volna azonban, ha ennek alapján B-t (önmagában) akár preprocesszornak, akár posztprocesszornak neveznénk.

Ezzel szemben, ha valamit önmagában tekintettünk preprocesszornak vagy posztprocesszornak, megvizsgálhatjuk azt, hogy ezek hol (mi előtt, illetve mi után) vannak a feldolgozás rendjében. A preprocesszor nyilván a processzor előtt, a posztprocesszor pedig a processzor után jut szerephez. Így tehát előfeldolgozásról, feldolgozásról és utófeldolgozásról van szó. Az elő- és utófeldolgozás pedig, e felosztás szerint, különbözne a feldolgozástól, nem annak része volna. Ez azonban nyilván elvi lehetetlenség, annak ellenére, hogy a gyakorlat itt-ott még használja a műveletek ily módon való felosztását. A preprocesszor, a processzor és a posztprocesszor jelenthet hardvert is, és az előfeldolgozást, a feldolgozást és az utófeldolgozást vezérlő szoftvert is. (Mind a preprocesszort, mind pedig a posztprocesszort nélkülözhető kifejezéseknek tartjuk.)

11. Nem lehet, mert *sorozatoknak elemei, soroknak* pedig *tagjai* vannak. *(A sor egy sorozat elemeiből képzett összeg.)*

12. Bárki meggyőződhet róla, hogy a pénzérme „fejre" vagy „írásra" *esésének* valószínűsége nulla, „élre", peremre esésének valószínűsége pedig 1. „Fej" és „írás" állapotban való *megállapodásának* valószínűsége egyaránt majdnem 1/2. (A pénzérme „állva" maradásának valószínűsége ugyanis nem nulla, mert az érme hengeralkotón, – ha recézett – alkotókon is stabilan megállhat, ahogyan ez kisebb méretű, nem kopott érméknél elő-előfordul.)

A kocka lapra *esésének* valószínűsége nulla. Élre *esésének* valószínűsége szintén nulla. Csúcsra esésének valószínűsége 1. Adott csúcsra *esésének* valószínűsége 1/8. Csúcson és élen való *megállapodásának* valószínűsége nulla, lapon való *megállapodásának* valószínűsége 1, adott lapon való *megállapodásának* valószínűsége pedig 1/6.

13. Nem, mert *az aritmetika valódi része a matematikának*. Egy híres professzor – úgy mesélik – azzal a megszólítással kezdte előadásait, hogy „Uraim és vaskohászok!" Ezzel az általunk nagyrabecsült vaskohászokat – nyilván tréfából – kizárta az urak közül. Mi ne zárjuk ki az aritmetikát a matematikából, és azt a másik gyakori hibát se kövessük el, amit a „geometria és matematika" osztályozást használók elkövetnek. (Tudnunk kell azonban, hogy e kategóriák jelentése többé-kevésbé változott az idők folyamán. A matematika például, nem is olyan régen, még a mérnöki, hadmérnöki tudományokat is jelentette.)

14. Nem. Mínusz kettő egészrésze mínusz kettő, ami nem kisebb mint mínusz kettő.

15. Tudomásunk szerint *eddig nem volt*. Természetes logaritmus alapszáma alapú hatvány viszont van. Ha ezt valaki hosszúnak találja kimondani, mondhat helyette *e* alapú hatványt, ahogyan ezt régóta az iskolákban is így tanítják. A természetes alapú hatvány után természetes lenne a természetes kitevőjű hatvány bevezetése is. Természetesen vetődne fel a kérdés, hogy melyik a természetesebb, a természetes alapú vagy a természetes kitevőjű hatvány? Nem tudnánk eldönteni. Abban természetesen mindenki egyetértene, hogy a legtermészetesebb hatvány az, amelynek alapja is és kitevője is természetes. Ennek a legtermészetesebb hatványnak az értéke azonban természetesen nem volna természetes. Látható ebből, hogy a túlzott természetesség mennyire természetellenes lehet.

Mivel a *természetes* nem tekinthető egzakt fogalomnak, a természetes logaritmus kifejezés sem természetes, hiszen így a 10-es alapú logaritmus egy volna a „nemtermészetes logaritmusok" közül. Az *e* alapú logaritmus szakkifejezést egzaktabbnak tartjuk, mint a természetes logaritmust.

16. A hiba az, hogy a magyar szóhasználat szerint *nem a kitevőt emeljük az alapra, hanem az alapot a kitevőre*. (Nemcsak azért, mert a kitevő szokott feljebb lenni, és feljebb levőt nem lehet lejjebb levőre „emelni".)

17. Nem igaz. A tangensérték inverze és az inverztangens értéke nem ugyanaz. A magyarban az inverztangens (azaz a tangens inverzfüggvénye) helyett az arcustangens (illetve, ha a főértékről van *szó*, akkor az Arcustangens) szó az elfogadott. Ha inverztangens értéket, vagy magyarul, a tangens inverzfüggvényének értékét számítjuk ki, arcustangens függvényt használunk. Ha tangensérték inverzét számítjuk ki, akkor először egy tangensértéket számítunk ki, majd ennek vesszük az inverzét. Mivel a tangensérték valós szám, a valós számoknak pedig a nulla kivételével két inverze is van, a „tangensérték inverze" nem egyértelmű. (Egy valós szám additív inverze a szám ellentettje, multiplikatív inverze pedig – ha a szám nem nulla – a szám reciproka. Az arcustangens esetében nyilván egyikről sincs szó.)

18. Nem. A rossz helyre írt számjegy *lehet* szintaktikus hiba is, és szemantikus hiba is. A helytelen műveleti utasításra ugyanez áll. A *helytelenül írt* műveleti utasítás azonban biztosan szintaktikus hiba. A helytelen műveleti utasítás és helytelenül írt műveleti utasítás *meg nem különböztetése és összetévesztése* pedig *biztosan szemantikus hiba*, még akkor is, ha nem is programban, hanem „csak" szövegben szerepel.

19. Nem. Forrásprogramnak valamilyen programokon manipuláló eljárás bemeneti programját nevezzük. Egy konkrét esetben hexadecimális számjegyekkel felírt program is lehet forrásprogram, de nem minden forrásprogram hexadecimális számjegyekkel felírt program. Lehetséges, hogy egy programokkal dolgozó eljárás több forrásprogrammal dolgozik, és az egyik 16-os alapú, egy másik pedig 2-es alapú számábrázolási rendszerben áll rendelkezésére. A definíció szerint csak a „hexadecimális számjegyekkel felírt" program lehetne forrásprogram.

A forrásprogram fogalompárja az ún. tárgyprogram. A forrásprogramok operandusprogramjai, a tárgyprogramok pedig operátumprogramjai (működésének eredményei) egy programokkal dolgozó eljárásnak.

20. Véleményünk szerint nem, mert minden nyelv – akár számítástechnikai, akár nem – csak szimbolikus nyelv lehet, azaz olyan nyelv, amely jeleket használ. Vagy van közlés jelek használata nélkül is?

Vasúti helyfoglalás

A frankfurti helyelosztó központ számítógépes nyilvántartásában Európa szinte valamennyi vasúttársaságának személy- és hálókocsiparkja szerepel. A közös munkába a szocialista országok közül eddig csak Magyarország kapcsolódott be. Ennek a központnak segítségével már az utazás előtt 90 nappal igényelhetők a nemzetközi vonatokra szóló helyjegyek. A MÁV mintegy 1000 jegypénztáránál, illetve az utazási irodákban a hazánkból induló nemzetközi expresszvonatokra azonnal, a máshonnan induló vonatokra a lekérési idő figyelembevételével szolgálja ki utasait. A lekérési idő attól függ, hogy milyen hírközlő berendezésen – telex, vasúti géptávíró, telefon, esetleg az erre a célra rendszeresített nyomtatvány – történik a jegyigénylés.

A magyar nyelv szótára

A Magyar Tudományos Akadémia elnöksége határozatot hozott egy új elvekre épülő számítógépes nagyszótár összeállítására. A magyar szókészlet e kincsesházában 10 millió szó és annak szövegkörnyezete kap helyet, természetesen a pontos forrásmegjelöléssel együtt. A szakértők ehhez a magyar könyvnyomtatás kezdetétől napjainkig vesznek mintát a nyomtatott emlékekből.

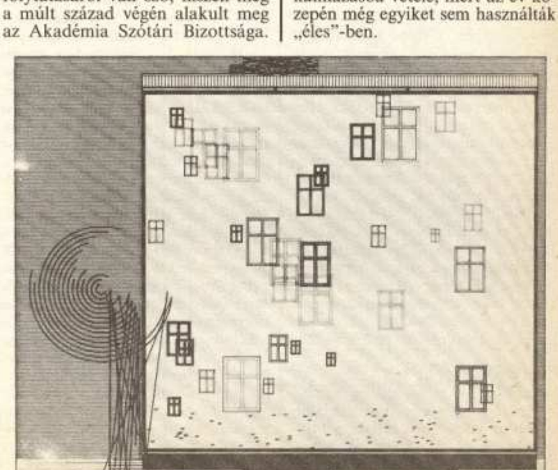
Nagy hagyományú munka folytatásáról van szó, hiszen még a múlt század végén alakult meg az Akadémia Szótári Bizottsága. Az volt az elgondolás, hogy a felvilágosodás korától kezdve kell összeválogatni a magyar próza, vers és nyomtatott sajtó szókincsét. Ma már több száz folyóméternyi cédulán 5 millió adat van feljegyezve több százezer magyar szóról.

A számítógép alkalmazása új lehetőségeket kínál a kutatók számára, mivel a gépi adathordozón levő anyag számtalan változatban visszakérdezhető. Készülhet írói szótár az egyes írókra legjellemzőbb szavakról, kifejezésekről, jelentős segítséget kaphatnak a költészeti hangtannal kapcsolatos kutatások is. A szótár létrehozása lehetővé teszi nyelvünk fejlődésének tudatosabb követését, sőt befolyásolását is.

Fókuszban a lokális hálózatok

Az idei évet a hazai számítástechnikában a helyi hálózatok megjelenése évének is tekinthetjük. Viszonylag nagy számú, 8–10 ezer darab, elosztott feladatú mikroszámítógépet helyi átvitellel ún. lokális hálózatba kapcsolva lehetővé válik egy-egy iroda, üzem, áruház teljes tevékenységének számítógépes kezelése, követése és irányítása.

Az idén megjelent, illetve megjelenő hazai kínálat igen gazdag. Az MTA KFKI a LOCHNESS rendszert ajánlja, az MTA SZTAKI a Cobust, a Medicor az MMT-HNS-t, a Mikropo a Mirealt (kompatíbilis az MP/M-mel), az SZKI a Propnet és a Propmult rendszert, a Videoton pedig az Exlocot. Komoly feladatot jelent e hardver- és szoftvertermékek alkalmazásba vétele, mert az év közepén még egyiket sem használták „éles"-ben.



Az új hordozható Erika

Az NDK-beli robotron kombinát elektronikus, margarétakerekes írógépei egyre népszerűbbek hazánkban. A Z80 mikroprocesszor funkcionális megfelelőjével vezérelt írógépekből ugyanis már az idén 5000 darabot importáltunk. Ezek jelentőségét tovább növeli, hogy illesztővel nyomtatóként mikroszámítógépekhez is csatlakoztathatók.

A gyártó tovább bővíti elektronikus írógépcsaládját. Az idei év újdonsága a hordozható Erika elektronikus írógép, amely az S 6005 kódszámot viseli. A hagyományos villanyírógépek emelgetéséhez szokott olvasókat bizonyára meglepi, hogy a nagykocsis gép súlya mindössze 10 kp.

A jövő áruháza

A jövő bevásárlóközpontjának modelljét építette fel a Seibu japán áruházi konszern 30 millió dolláros költséggel a Tokió melletti tudósvárosban, Cukubában. Egybegyűjtötték itt a japán vállalatoktól a technika összes elképzelhető vívmányát, hogy kipuhatolják: mely műszaki megoldások fogják jellemezni a jövő áruházát, melyek nyerik meg a vásárlók tetszését vagy bizonyulnak életképtelennek.

Egy kis ízelítő a jövő áruházának működéséből: a vásárolt árut nem kell a kosarunkban vinni vagy kiskocsin tolni, hanem csomagcipelő robotok követik a vevők derekára helyezett mini adóvevő utasításait. A tájékozatlan vásárlónak számítógépek adnak felvilágosítást az óhajtott cikkek helyéről, de igénybe lehet venni az áruház tévéinformátorát is, amelyen előzékeny alkalmazott ad felvilágosítást. A választás nehézségeivel elfoglalt szülők ugyancsak tévéképernyőn kísérhetik figyelemmel, hogy tölti idejét csemetéjük a gyermekmegőrzőben vagy az új, elektronikus játékok kipróbálására szolgáló játszószobában.

A cukubai áruház valószínűleg a hölgyvásárlók paradicsoma lesz: ruhaszalonjában számítógépek segítenek az egyéniséghez illő ruhaanyag és színösszeállítás megválasztásában. A szupermodern próbafülkében pedig lézerberendezés varázsolja a vevő mögé azt a környezetet (például esküvői ünnepséget, házibulit, teniszpályát), amelyben a vásárolni kívánt ruhát viselni fogja.

Unix kompatibilis Commodore gép

A júniusi Consumer Electronics Show-n mutatták be először az új Commodore gyártmányválaszték legnagyobb teljesítményű tagját. A C900 típusszámot viselő modell alapgépe 16 Mbájt címzési képességű, nagy teljesítményű 16 bites processzoron alapszik. A kiépíthető memóriakapacitás 0,5–2 Mbájt, a Winchester lemeztároló kapacitása pedig 20–67 Mbájt. Az Unix kompatibilis Coherent operációs rendszert alkalmazó gép maximum 8 felhasználót tud kiszolgálni. Nagy teljesítményű grafikus munkaállomás konfigurációban is szállítani kívánják. A 128 kbájtos saját memóriával és általános grafikai alapprimitivát (bit block transfer) megvalósító chippel rendelkező grafikus terminál a csatolható egér-periféria segítségével több ablakos üzemmódban is használható. Az új gép 1985 végén kerül az üzletekbe.

Elektronizált étterem

A kaliforniai Paloalte éttermeknek minden asztalába terminált építettek. A pincér ezen billentyűzi be a rendelést, amit a konyhában levő terminál azonnal megjelenít. Így mindjárt hozzálátnak az étel elkészítéséhez, és a fogások elkészültét a konyha a saját terminálja segítségével tudja visszajelezni. A számlát a vendégek asztalán levő terminál nyomtatója adja.

Az étterem vezetője szerint a vendégeknek tetszik az ilyen mértékben elektronizált étterem. Előnye még, hogy egyszerűbb az irányítás is, hiszen a vezető kényelmesen, folyamatosan figyelemmel kísérheti az étterem valamennyi dolgozójának munkáját és az összes asztal kiszolgálását. Az IBM számítógéppel működő rendszernek egyetlen árnyoldala, hogy most még meglehetősen drága, és használatát csak azok az éttermek engedhetik meg maguknak, amelyek legalább háromnegyed millió dolláros évi forgalmat bonyolítanak le.

Változóban a PC piac

A Commodore 1985. július 24-én jelentette be piaci forgalmazásra a korábban felvásárolt Amiga cég második generációs személyi számítógépét. Az Amiga gép 7,8 MHz-es, Motorola 68000-es mikroprocesszoron alapszik, 256 kbájt RAM alapkapacitású és 640 × 400 képpont felbontásban, 16 színben képes grafikát megjeleníteni. A 880 kbájt kapacitású, 3 1/2 hüvelykes floppytárolóval felszerelt berendezés az AmigaDOS-nak nevezett, gyártóspecifikus operációs rendszert használja, de egy külön emulációs szoftver támogatásával az IBM PC DOS alatt működő szoftverek futtatására is képes. A konstrukció nyílttá tételével a Commodore egyúttal ösztönözni próbálja a szélesebb körű, saját szoftverellátást is. A számítógépes szaküzletekben árusított új PC listára 1295 dollár, a színes monitor pedig 495 dollárba kerül. Az első évben 200 ezer gép eladását tervezik.

A Commodore volt elnöke, Jack Tramiel által irányított Atari is piacra lépett új, második generációs termékével. Az 52OST modell szintén a Motorola 68000-es processzoron alapszik, RAM alapkapacitása 512 kbájt, és 512 kbájtos floppytárolóval van felszerelve. A Digital Research Macintosh-szerű, GEMDOS operációs rendszerét használó új gépet szuperalacsony áron hozták ki. A 12 hüvelykes képátlójú, monokromatikus monitorral felszerelt teljes konfiguráció ára mindössze 800 dollár.

A második generációs, grafikaorientált személyi számítógépek piacán ádáz küzdelem van kibontakozóban. Az eddig monopolhelyzetben lévő Apple válasza sem késlekedhet sokáig (jelentős árcsökkentés a régi Macintosh-ra és növelt teljesítményű, új Mac megjelenése?). A régi PC konstrukciók eladási nehézségeit komoly árcsökkentéssel és újabb hardver-, valamint szoftvereszközökkel igyekeznek ellensúlyozni a gyártók. A Commodore 64 további árcsökkentése mellett egy 600 dolláros, 10 Mbájt kapacitású Winchester lemeztároló piacra hozatalával akarja például a Commodore kelendőbbé tenni régi termékét, és csökkenteni jelentős készleteit. Az IBM is aligha tud másképpen szabadulni 500 ezret meghaladó készletétől. Az ipari pletykák világában PC2-nek nevezett, második generációs gépe piacra hozatalával is csak azért vár, hogy ki tudja árusítani régi termékeit.

Kedves Olvasónk!

Szíves tájékoztatására közöljük a Lapkiadó Vállalat azon budapesti és vidéki üzleteinek címét, ahol lapunkat, a Mikroszámítógép Magazint megvásárolhatja:

BUDAPEST

VII., Almássy tér 11.
VII., Dohány u. 82.
VII., Dohány u. 84.
IX., Üllői út 119.
XII., Böszörményi út 3/a.
XIII., Rajk László u. 24.
XIV., Thököly út 21.

VIDÉK

Bükfürdő, pavilonsor
Debrecen, Simonfi u. 1.
Győr, Árpád út 63.
Kőszeg, Jurisics tér 10.
Sopron, Orsolya tér, Lábasház
Szeged, Bajcsy-Zsilinszky u. 19.
Szombathely, Berzsenyi Dániel tér 3.
Pécs, Jókai u. 25.

M08X

proper 8 **proper 16**

PROFESSZIONÁLIS SZEMÉLYI SZÁMÍTÓGÉP

3 ÉVE AZ ALKALMAZÓK MUNKAESZKÖZE

(Vállalat–Intézmény–Állami Gazdaság–MGTSZ)

NAGY TELJESÍTMÉNYŰ MUNKAHELY A MINDENKORI FELADATHOZ ILLESZTVE

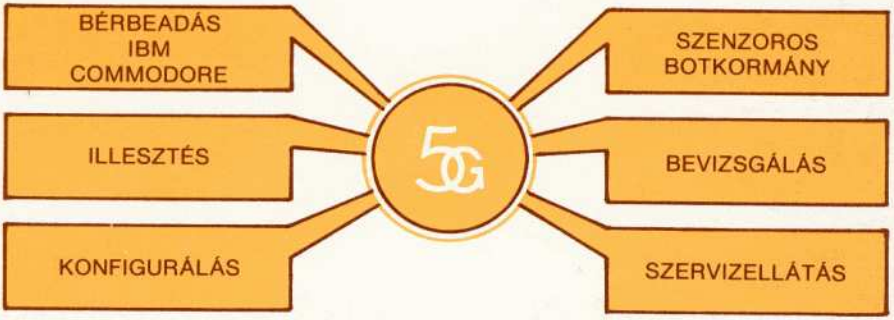
- Adatfeldolgozás, ügyvitel, szövegszerkesztés
- Irodai szövegfeldolgozás – levélminőségű magyar karakterkészletű nyomtatóval
- Mérnöki tervezés, műszaki számítás, grafika
- Széles körű szolgáltatás – terminál emuláció, adatbáziskezelés

VÁLLALJUK
- Bemutatók szervezését
- Kulcsrakész rendszerek kialakítását
- A szoftverek egyéni kipróbálását saját gépen

SW '86

TÁJÉKOZTATÁST AD:
Szki Sci-L
Rendszerértékesítő Iroda
Budapest, I., Iskola u. 10. 1011
Telefon: 260-000

SW.'86



5G

SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÓ KISSZÖVETKEZET

1067 Budapest, VI., Csengery u. 53.
Tel.: 321-339/170, 171 Telex: 22-7392 5746

ÁRA: 30,– FORINT

HOTEL DUNA INTER•CONTINENTAL

1985. november 12–15.

Mi is
várjuk Önt!

1024 Budapest II., Petrezselyem utca 6.

SZOFTVERKERESKEDELMI ÉS FEJLESZTÉSI BETÉTI TÁRSULÁS

Telefon: 358-530/866 és 499-es mellék